

DDDDDDDDDDDDDD	CCCCCCCCCCCC	LLL
DDDDDDDDDDDDDD	CCCCCCCCCCCC	LLL
DDDDDDDDDDDDDD	CCCCCCCCCCCC	LLL
DDD	DDD CCC	LLL
DDDDDDDDDDDDDD	CCCCCCCCCCCC	LLLLLLLLLLLL
DDDDDDDDDDDDDD	CCCCCCCCCCCC	LLLLLLLLLLLL
DDDDDDDDDDDDDD	CCCCCCCCCCCC	LLLLLLLLLLLL

\*\*FILE\*\*ID\*\*COMMAND

C 7

CCCCCCCC CCCCCCCC  
CCCCCCCC 000000 000000 MM MM MM MM MM MM  
CC 00 00 MMMM MMMM MMMM MMMM MM AA AA NN NN NN DDDDDDDD  
CC 00 00 MMMM MMMM MMMM MMMM MM AA AA NN NN NN DD DD  
CC 00 00 MM MM MM MM MM MM AA AA AA NNNN NN DD DD  
CC 00 00 MM MM MM MM MM MM AA AA AA NNNN NN DD DD  
CC 00 00 MM MM MM MM MM MM AA AA AA NN NN NN DD DD  
CC 00 00 MM MM MM MM MM MM AA AA AA NN NN NN DD DD  
CC 00 00 MM MM MM MM MM MM AA AA AA NN NN NN DD DD  
CC 00 00 MM MM MM MM MM MM AA AA AA NN NN NN DD DD  
CC 00 00 MM MM MM MM MM MM AA AA AA NN NN NN DD DD  
CC 00 00 MM MM MM MM MM MM AA AA AA NN NN NN DD DD  
CCCCCCCC 000000 000000 MM MM MM MM MM AA AA NN NN NN DDDDDDDD  
CCCCCCCC

....  
....  
....

LL IIIII SSSSSSS  
LL IIIII SSSSSSS  
LL II SSS  
LLLLLLLL LLLLIII SSSSSSS  
LLLLLLLL LLLLIII SSSSSSS

(3)	141	PROCESS NEXT COMMAND
(4)	170	PROCESS REST OF COMMAND
(6)	653	CALL LOGINOUT TO ABORT THE PROCESS
(7)	683	EOD/DECK COMMANDS
(8)	703	CHECK FOR CONTROL Y/C AST PENDING
(9)	724	ENABLE/DISABLE CONTROL Y/C AST'S
(10)	758	FLUSH COMMAND BUFFER
(11)	788	PROCESS FOREIGN COMMAND
(12)	831	GET INTERNAL ROUTINE INDEX

0000 1 :TITLE COMMAND - PROCESS NEXT COMMAND  
0000 2 :IDENT 'V04-000'  
0000 3 \*\*\*\*\*  
0000 4 \*  
0000 5 \* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY  
0000 6 \* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.  
0000 7 \* ALL RIGHTS RESERVED.  
0000 8 \*  
0000 9 \*  
0000 10 \* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
0000 11 \* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
0000 12 \* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
0000 13 \* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
0000 14 \* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
0000 15 \* TRANSFERRED.  
0000 16 \*  
0000 17 \*  
0000 18 \* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
0000 19 \* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
0000 20 \* CORPORATION.  
0000 21 \*  
0000 22 \* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
0000 23 \* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.  
0000 24 \*  
0000 25 \*  
0000 26 \*\*\*\*\*  
0000 27  
0000 28 :PROCESS NEXT COMMAND  
0000 29  
0000 30 D. N. CUTLER 22-MAR-77  
0000 31  
0000 32 MODIFIED BY:  
0000 33  
0000 34 V03-018 CWH3018 CW Hobbs 28-Jul-1984  
0000 35 Fix HWS0093 to avoid loop on a control/y.  
0000 36  
0000 37 V03-017 HWS0093 Harold Schultz 23-Jul-1984  
0000 38 If interactive and at indirect level zero, unconditionally  
0000 39 clear the execute-only procedure flag.  
0000 40  
0000 41 V03-016 HWS0061 Harold Schultz 19-Apr-1984  
0000 42 Restore locked keypad state only after PRC\_V\_CNTRLY  
0000 43 is cleared.  
0000 44  
0000 45 V03-015 HWS0020 Harold Schultz 06-Mar-1984  
0000 46 When parsing a command which begins with a verb, ignore  
0000 47 leading blanks when expanding a symbol definition.  
0000 48  
0000 49 V03-014 HWS0008 Harold Schultz 13-Feb-1984  
0000 50 Remove obsolete code for setting up prompt block in PRC.  
0000 51  
0000 52 V03-013 PCG0015 Peter George 06-Feb-1984  
0000 53 Fix EOJ so that it works even if LOGOUT is deleted.  
0000 54 Fix bug in force exiting a privileged image.  
0000 55  
0000 56 V03-012 PCG0014 Peter George 12-Jan-1984  
0000 57 Fix typo.

0000	58	
0000	59	V03-011 PCG0013 Peter George Set FLUSH bit in flush routine. 03-Jan-1984
0000	60	
0000	61	
0000	62	V03-010 PCG0012 Peter George Disable indirection when looking ahead for null nodes. 16-Aug-1983
0000	63	
0000	64	
0000	65	V03-009 PCG0011 Peter George Support null node names. 20-Jul-1983
0000	66	
0000	67	
0000	68	V03-008 PCG0010 Peter George Fix broken branch. 15-Jun-1983
0000	69	Remove extraneous labels.
0000	70	Remove references to PRC_L_TAB_VEC.
0000	71	
0000	72	
0000	73	V03-007 PCG0009 Peter George Have command dispatch table branch to JMP's. 27-May-1983
0000	74	
0000	75	
0000	76	V03-006 PCG0008 Peter George Move token sorting routine to DESCVAL. 30-Apr-1983
0000	77	Clear WRK_B_PARMSUM.
0000	78	
0000	79	
0000	80	V03-005 PCG0007 Peter George Move GOTO scanning to GOTO module. 01-Apr-1983
0000	81	
0000	82	
0000	83	V03-004 PCG0006 Peter George Update to new structure level. 15-Feb-1983
0000	84	Remove SETBIT WRK_V_NOSTAT in EOD.
0000	85	Clear PTR_L_ENTITY and PTR_B_NUMBER when doing DCL\$GENDESCR.
0000	86	Sort new larger PTR blocks.
0000	87	Add DCL\$LOOKUP_INTERNAL.
0000	88	Init recall and keypad fields in WRK block.
0000	89	
0000	90	
0000	91	V03-003 PCG0005 Peter George Use WRK_C_SCRSTKSIZ and WRK prompt string descriptor. 16-Nov-1982
0000	92	
0000	93	
0000	94	V03-002 PCG0004 Peter George Use PRC data structure for prompt string. 28-Oct-1982
0000	95	
0000	96	
0000	97	V03-002 PCG0003 Peter George Fix batch logout accvio caused by DCL\$ABORT. 19-Oct-1982
0000	98	Use DCL\$ABORT for DCL\$EOJ.
0000	99	
0000	100	
0000	101	V03-001 PCG0002 Peter George Move common command parsing code to DCL\$PARSE_COMMAND 30-Sep-1982
0000	102	in PARSENT. Initialize WRK_L_PROMPTRN, WRK_B_VALLEV,
0000	103	and WRK_B_CMDOPT.
0000	104	
0000	105	--

0000 107 :  
0000 108 : MACRO LIBRARY CALLS  
0000 109 :  
0000 110 :  
0000 111 :SSCLITABDEF : DEFINE TABLE STRUCTURES  
0000 112 :PTRDEF : DEFINE RESULT PARSE DESCRIPTOR FORMAT  
0000 113 :SYMDEF : DEFINE SYMBOL ENTRY OFFSETS  
0000 114 :PRCDEF : DEFINE PROCESS WORK AREA  
0000 115 :WRKDEF : DEFINE COMMAND WORK AREA  
0000 116 :SCLIMSGDEF : DEFINE ERROR/STATUS VALUES  
0000 117 :SDEVDEF : DEFINE DEVICE CHARACTERISTIC BITS  
0000 118 :SPSLDEF : DEFINE PROCESSOR STATUS FIELDS  
0000 119 :SRABDEF : DEFINE RAB OFFSETS  
0000 120 :  
00000000 121 .PSECT DCL\$ZDATA,BYTE,RD,NOWRT  
54 55 4F 47 4F 4C 00' 0000 122 :  
06 0000 123 LOGOUT: .ASCIC 'LOGOUT'  
00000000 124 :  
0007 125 ROUTINES\_LEN = 0  
0007 126 :  
0007 127 .MACRO INTIMAGE NAME  
0007 128 \$NAM1\$ = ^A/%EXTRACT(0,4,NAME)/!^X80  
0007 129 \$NAM2\$ = ^A/%EXTRACT(4,4,NAME)/  
0007 130 .LONG \$NAM1\$  
0007 131 .LONG \$NAM2\$  
0007 132 ROUTINES\_LEN = ROUTINES\_LEN + 8  
0007 133 .ENDM  
0007 134 :  
0007 135 ROUTINES:  
0007 136 INTIMAGES ; GENERATE TABLE OF ROUTINE NAMES  
01B7 137 :  
00000000 138 .PSECT DCL\$ZCODE,BYTE,RD,NOWRT  
0000 139 :  
0000 140 :  
0000 141 :  
0000 142 :  
0000 143 :  
0000 144 :  
0000 145 :  
0000 146 :  
0000 147 :  
0000 148 :  
0000 149 :  
0000 150 :  
0000 151 :  
0000 152 :  
0000 153 :  
0000 154 :  
0000 155 :  
0000 156 :  
0000 157 :  
0000 158 :  
0000 159 :  
0000 160 :  
0000 161 :  
0000 162 :  
0000 163 :  
0000 164 :  
0000 165 :  
0000 166 :  
0000 167 :  
0000 168 :  
0000 169 :  
0000 170 :  
0000 171 :  
0000 172 :  
0000 173 :  
0000 174 :  
0000 175 :  
0000 176 :  
0000 177 :  
0000 178 :  
0000 179 :  
0000 180 :  
0000 181 :  
0000 182 :  
0000 183 :  
0000 184 :  
0000 185 :  
0000 186 :  
0000 187 :  
0000 188 :  
0000 189 :  
0000 190 :  
0000 191 :  
0000 192 :  
0000 193 :  
0000 194 :  
0000 195 :  
0000 196 :  
0000 197 :  
0000 198 :  
0000 199 :  
0000 200 :  
0000 201 :  
0000 202 :  
0000 203 :  
0000 204 :  
0000 205 :  
0000 206 :  
0000 207 :  
0000 208 :  
0000 209 :  
0000 210 :  
0000 211 :  
0000 212 :  
0000 213 :  
0000 214 :  
0000 215 :  
0000 216 :  
0000 217 :  
0000 218 :  
0000 219 :  
0000 220 :  
0000 221 :  
0000 222 :  
0000 223 :  
0000 224 :  
0000 225 :  
0000 226 :  
0000 227 :  
0000 228 :  
0000 229 :  
0000 230 :  
0000 231 :  
0000 232 :  
0000 233 :  
0000 234 :  
0000 235 :  
0000 236 :  
0000 237 :  
0000 238 :  
0000 239 :  
0000 240 :  
0000 241 :  
0000 242 :  
0000 243 :  
0000 244 :  
0000 245 :  
0000 246 :  
0000 247 :  
0000 248 :  
0000 249 :  
0000 250 :  
0000 251 :  
0000 252 :  
0000 253 :  
0000 254 :  
0000 255 :  
0000 256 :  
0000 257 :  
0000 258 :  
0000 259 :  
0000 260 :  
0000 261 :  
0000 262 :  
0000 263 :  
0000 264 :  
0000 265 :  
0000 266 :  
0000 267 :  
0000 268 :  
0000 269 :  
0000 270 :  
0000 271 :  
0000 272 :  
0000 273 :  
0000 274 :  
0000 275 :  
0000 276 :  
0000 277 :  
0000 278 :  
0000 279 :  
0000 280 :  
0000 281 :  
0000 282 :  
0000 283 :  
0000 284 :  
0000 285 :  
0000 286 :  
0000 287 :  
0000 288 :  
0000 289 :  
0000 290 :  
0000 291 :  
0000 292 :  
0000 293 :  
0000 294 :  
0000 295 :  
0000 296 :  
0000 297 :  
0000 298 :  
0000 299 :  
0000 300 :  
0000 301 :  
0000 302 :  
0000 303 :  
0000 304 :  
0000 305 :  
0000 306 :  
0000 307 :  
0000 308 :  
0000 309 :  
0000 310 :  
0000 311 :  
0000 312 :  
0000 313 :  
0000 314 :  
0000 315 :  
0000 316 :  
0000 317 :  
0000 318 :  
0000 319 :  
0000 320 :  
0000 321 :  
0000 322 :  
0000 323 :  
0000 324 :  
0000 325 :  
0000 326 :  
0000 327 :  
0000 328 :  
0000 329 :  
0000 330 :  
0000 331 :  
0000 332 :  
0000 333 :  
0000 334 :  
0000 335 :  
0000 336 :  
0000 337 :  
0000 338 :  
0000 339 :  
0000 340 :  
0000 341 :  
0000 342 :  
0000 343 :  
0000 344 :  
0000 345 :  
0000 346 :  
0000 347 :  
0000 348 :  
0000 349 :  
0000 350 :  
0000 351 :  
0000 352 :  
0000 353 :  
0000 354 :  
0000 355 :  
0000 356 :  
0000 357 :  
0000 358 :  
0000 359 :  
0000 360 :  
0000 361 :  
0000 362 :  
0000 363 :  
0000 364 :  
0000 365 :  
0000 366 :  
0000 367 :  
0000 368 :  
0000 369 :  
0000 370 :  
0000 371 :  
0000 372 :  
0000 373 :  
0000 374 :  
0000 375 :  
0000 376 :  
0000 377 :  
0000 378 :  
0000 379 :  
0000 380 :  
0000 381 :  
0000 382 :  
0000 383 :  
0000 384 :  
0000 385 :  
0000 386 :  
0000 387 :  
0000 388 :  
0000 389 :  
0000 390 :  
0000 391 :  
0000 392 :  
0000 393 :  
0000 394 :  
0000 395 :  
0000 396 :  
0000 397 :  
0000 398 :  
0000 399 :  
0000 400 :  
0000 401 :  
0000 402 :  
0000 403 :  
0000 404 :  
0000 405 :  
0000 406 :  
0000 407 :  
0000 408 :  
0000 409 :  
0000 410 :  
0000 411 :  
0000 412 :  
0000 413 :  
0000 414 :  
0000 415 :  
0000 416 :  
0000 417 :  
0000 418 :  
0000 419 :  
0000 420 :  
0000 421 :  
0000 422 :  
0000 423 :  
0000 424 :  
0000 425 :  
0000 426 :  
0000 427 :  
0000 428 :  
0000 429 :  
0000 430 :  
0000 431 :  
0000 432 :  
0000 433 :  
0000 434 :  
0000 435 :  
0000 436 :  
0000 437 :  
0000 438 :  
0000 439 :  
0000 440 :  
0000 441 :  
0000 442 :  
0000 443 :  
0000 444 :  
0000 445 :  
0000 446 :  
0000 447 :  
0000 448 :  
0000 449 :  
0000 450 :  
0000 451 :  
0000 452 :  
0000 453 :  
0000 454 :  
0000 455 :  
0000 456 :  
0000 457 :  
0000 458 :  
0000 459 :  
0000 460 :  
0000 461 :  
0000 462 :  
0000 463 :  
0000 464 :  
0000 465 :  
0000 466 :  
0000 467 :  
0000 468 :  
0000 469 :  
0000 470 :  
0000 471 :  
0000 472 :  
0000 473 :  
0000 474 :  
0000 475 :  
0000 476 :  
0000 477 :  
0000 478 :  
0000 479 :  
0000 480 :  
0000 481 :  
0000 482 :  
0000 483 :  
0000 484 :  
0000 485 :  
0000 486 :  
0000 487 :  
0000 488 :  
0000 489 :  
0000 490 :  
0000 491 :  
0000 492 :  
0000 493 :  
0000 494 :  
0000 495 :  
0000 496 :  
0000 497 :  
0000 498 :  
0000 499 :  
0000 500 :  
0000 501 :  
0000 502 :  
0000 503 :  
0000 504 :  
0000 505 :  
0000 506 :  
0000 507 :  
0000 508 :  
0000 509 :  
0000 510 :  
0000 511 :  
0000 512 :  
0000 513 :  
0000 514 :  
0000 515 :  
0000 516 :  
0000 517 :  
0000 518 :  
0000 519 :  
0000 520 :  
0000 521 :  
0000 522 :  
0000 523 :  
0000 524 :  
0000 525 :  
0000 526 :  
0000 527 :  
0000 528 :  
0000 529 :  
0000 530 :  
0000 531 :  
0000 532 :  
0000 533 :  
0000 534 :  
0000 535 :  
0000 536 :  
0000 537 :  
0000 538 :  
0000 539 :  
0000 540 :  
0000 541 :  
0000 542 :  
0000 543 :  
0000 544 :  
0000 545 :  
0000 546 :  
0000 547 :  
0000 548 :  
0000 549 :  
0000 550 :  
0000 551 :  
0000 552 :  
0000 553 :  
0000 554 :  
0000 555 :  
0000 556 :  
0000 557 :  
0000 558 :  
0000 559 :  
0000 560 :  
0000 561 :  
0000 562 :  
0000 563 :  
0000 564 :  
0000 565 :  
0000 566 :  
0000 567 :  
0000 568 :  
0000 569 :  
0000 570 :  
0000 571 :  
0000 572 :  
0000 573 :  
0000 574 :  
0000 575 :  
0000 576 :  
0000 577 :  
0000 578 :  
0000 579 :  
0000 580 :  
0000 581 :  
0000 582 :  
0000 583 :  
0000 584 :  
0000 585 :  
0000 586 :  
0000 587 :  
0000 588 :  
0000 589 :  
0000 590 :  
0000 591 :  
0000 592 :  
0000 593 :  
0000 594 :  
0000 595 :  
0000 596 :  
0000 597 :  
0000 598 :  
0000 599 :  
0000 600 :  
0000 601 :  
0000 602 :  
0000 603 :  
0000 604 :  
0000 605 :  
0000 606 :  
0000 607 :  
0000 608 :  
0000 609 :  
0000 610 :  
0000 611 :  
0000 612 :  
0000 613 :  
0000 614 :  
0000 615 :  
0000 616 :  
0000 617 :  
0000 618 :  
0000 619 :  
0000 620 :  
0000 621 :  
0000 622 :  
0000 623 :  
0000 624 :  
0000 625 :  
0000 626 :  
0000 627 :  
0000 628 :  
0000 629 :  
0000 630 :  
0000 631 :  
0000 632 :  
0000 633 :  
0000 634 :  
0000 635 :  
0000 636 :  
0000 637 :  
0000 638 :  
0000 639 :  
0000 640 :  
0000 641 :  
0000 642 :  
0000 643 :  
0000 644 :  
0000 645 :  
0000 646 :  
0000 647 :  
0000 648 :  
0000 649 :  
0000 650 :  
0000 651 :  
0000 652 :  
0000 653 :  
0000 654 :  
0000 655 :  
0000 656 :  
0000 657 :  
0000 658 :  
0000 659 :  
0000 660 :  
0000 661 :  
0000 662 :  
0000 663 :  
0000 664 :  
0000 665 :  
0000 666 :  
0000 667 :  
0000 668 :  
0000 669 :  
0000 670 :  
0000 671 :  
0000 672 :  
0000 673 :  
0000 674 :  
0000 675 :  
0000 676 :  
0000 677 :  
0000 678 :  
0000 679 :  
0000 680 :  
0000 681 :  
0000 682 :  
0000 683 :  
0000 684 :  
0000 685 :  
0000 686 :  
0000 687 :  
0000 688 :  
0000 689 :  
0000 690 :  
0000 691 :  
0000 692 :  
0000 693 :  
0000 694 :  
0000 695 :  
0000 696 :  
0000 697 :  
0000 698 :  
0000 699 :  
0000 700 :  
0000 701 :  
0000 702 :  
0000 703 :  
0000 704 :  
0000 705 :  
0000 706 :  
0000 707 :  
0000 708 :  
0000 709 :  
0000 710 :  
0000 711 :<br

0000 141 .SBTTL PROCESS NEXT COMMAND  
0000 142 + DCLSRESTART - PROCESS NEXT COMMAND  
0000 143 : THIS ROUTINE IS THE CENTRAL COMMAND PARSE AND DISPATCH ROUTINE FOR THE  
0000 144 : STARLET DCLS COMMAND LANGUAGE INTERPRETER. IT INITIALIZES FOR THE NEXT  
0000 145 : COMMAND, READS AND PARSES THE COMMAND FOR SYNTACTIC CORRECTNESS, AND THEN  
0000 146 : EXECUTES THE APPROPRIATE IMAGE.  
0000 147 :  
0000 148 :  
0000 149 :  
0000 150 :  
0000 151 :  
0000 152 : FP = ADDRESS OF WRK STRUCTURE  
0000 153 : R11 = ADDRESS OF PRC STRUCTURE  
0000 154 :  
0000 155 :  
0000 156 :  
0000 157 : COMMAND IS READ, PARSED, AND AN IMAGE IS ACTIVATED.  
0000 158 :  
0000 159 : .ENABL LSB  
0000 160 :  
0000 161 DCLSRESTART:: :  
5E F486 CD 9E 0000 162 MOVAB WRK\_K\_LENGTH(FP),SP :RESTART ENTRY POINT  
5A 5D D0 0005 163 :ALLOCATE COMMAND IMPURE AREA  
FFFF'CF 9E 0008 164 MOVL FP,R10 : AND RESET STACK POINTER  
F48E CA 000C 165 MOVAB W^DCLSHYPHEN-1,- :SET BASE ADDRESS OF WRK  
08 11 000F 166 WRK\_L\_CHARPTR(R10) :SET TO FORCE INPUT ON NEXT GET  
0011 167 BRB 1S :PROCESS COMMAND  
0011 168

```

0011 170 .SBTTL PROCESS REST OF COMMAND
0011 171 :+ DCLSCMDSTART - PROCESS REST OF COMMAND
0011 172 : THIS ROUTINE IS THE CENTRAL COMMAND PARSE AND DISPATCH ROUTINE FOR THE
0011 173 : STARLET DCLS COMMAND LANGUAGE INTERPRETER. IT INITIALIZES FOR THE COMPLETION
0011 174 : OF THE CURRENT COMMAND LINE, READS AND PARSES THE COMMAND FOR SYNTACTIC
0011 175 : CORRECTNESS, AND THEN EXECUTES THE APPROPRIATE IMAGE.
0011 176 : INPUTS:
0011 177 : 180 :
0011 181 :     FP = ADDRESS OF WRK STRUCTURE
0011 182 :     R11 = ADDRESS OF PRC STRUCTURE
0011 183 :     WRK_L_CHARPTR(FP) = POINTER INTO COMMAND INPUT BUFFER
0011 184 :     WRK_G_BUFFER(FP) = COMMAND EXPANSION BUFFER
0011 185 :     WRK_G_INPBUF(FP) = COMMAND INPUT BUFFER
0011 186 : 187 : OUTPUTS:
0011 188 : 189 :     COMMAND IS READ, PARSED, AND AN IMAGE IS ACTIVATED.
0011 190 :-
0011 191 : 192 DCLSCMDSTART:: :COMMAND PROCESSING ENTRY POINT
0011 193 : 194 : ALLOCATE AND INITIALIZE COMMAND SCRATCH WORK AREA
0011 195 : 196 : MOVAB WRK_K_LENGTH(FP),SP :ALLOCATE COMMAND IMPURE AREA
0011 197 : 198 : MOVL FP,R10 :AND RESET STACK POINTER
0011 199 : 199 : MOVZBW PRC_B_PROMPTLEN(R11),- :SET BASE ADDRESS OF WRK
0011 200 : 200 : WRK_W_PROMPTLEN(R10) :SET PROMPT LENGTH
0011 201 : 201 : MOVAB PRC_W_PROMPTCTRL(R11),- :SET PROMPT ADDRESS
0011 202 : 202 : WRK_L_PROMPTADDR(R10) :SET REGULAR PROMPT
0011 203 : 203 : CLRB PRC_B_CONTINUE(R11) :ARE WE AT COMMAND LEVEL 0?
0011 204 : 204 : TSTL PRC_L_INDEPTH(R11) :NO, CHECK CRTL-Y FLAGS
0011 205 : 205 : BNEQ 28
0011 206 : 207 : BBS #PRC_V_MODE,PRC_W_FLAGS(R11),5$ :SKIP IF BATCH MODE
0011 208 : 208 : CLRB PRC_B_EXONLYL(R11) :CLEAR EX-ONLY FLAG
0011 209 : 209 : BRB 5$ :USE 'S' PROMPT
0011 210 : 210 : 28: BBS #PRC_V_YLEVEL,PRC_W_FLAGS(R11),5$ :USE 'S' IF CONTROL/Y LEVEL
0011 211 : 211 : BBS #PRC_V_CNTRLY,PRC_W_FLAGS(R11),5$ : (ALSO IF CTRL/Y JUST HIT)
0011 212 : 212 : MOVB #^A/-,PRC_B_CONTINUE(R11) :NO -- PROMPT STRING = 'S'
0011 213 : 213 : 55: MOVAB WRK_G_BUFFER(R10),WRK_L_EXPANDPTR(R10) :SET ADDRESS OF EXPANSION BUF
0011 214 : 214 : MOVAB WRK_G_RESULT(R10),WRK_L_RSLNXT(R10) :SET ADDRESS OF RESULT PARSE TAB
0011 215 : 215 : MOVAB DCLSSINPUT,WRK_L_READRTN(R10) :SET RECORD INPUT ROUTINE
0011 216 : 216 : MOVAB DCLSSINPUT,WRK_L_PROMPTRTN(R10) :SET PROMPT ROUTINE
0011 217 : 217 : MOVAB DCLSSPECIAL,WRK_L_SPECRTN(R10) :SET SPECIAL PROCESSING ROUTINE
0011 218 : 218 : MOVAB DCLSCHARERRR,WRK_L_ERRORRTN(R10) :SET ERROR HANDLER ROUTINE
0011 219 : 219 : MOVAB DCLSSERRMSG,WRK_L_SIGNALRTN(R10) :SET ERROR SIGNALING ROUTINE
0011 220 : 220 : CLRW WRK_W_FLAGS(R10) :RESET COMMAND FLAGS
0011 221 : 221 : CLRBL WRK_B_VALLEV(R10) :RESET VALUE LEVEL
0011 222 : 222 : CLRBL WRK_B_CMDOPT(R10) :RESET OPTION NUMBER
0011 223 : 223 : CLRBL WRK_L_RSLEND(R10) :ZERO ADDR OF LAST TOKEN DESC
0011 224 : 224 : CLRBL WRK_L_QUABLK(R10) :ZERO QUABLK
0011 225 : 225 : CLRBL WRK_L_PROPTR(R10) :ZERO PROPTR
0011 226 : 226 : CLRBL WRK_L_IMAGE(R10) :ZERO ADDRESS OF IMAGE NAME

```

COMMAND  
V04-000

- PROCESS NEXT COMMAND  
PROCESS REST OF COMMAND

J 7

15-SEP-1984 23:40:03  
4-SEP-1984 23:39:43

15-SEP-1984 23:40:03 VAX/VMS Macro V04-00  
4-SEP-1984 23:39:43 [DCL.SRC]COMMAND.MAR:1

Page 6  
(4)

D2 AA	D4	0099	227	CLRL	WRK_L_PAROUT(R10)	;	
D1 AA	94	009C	228	CLRB	WRK_B_MINPARM(R10)	;	
DO AA	94	009F	229	CLRB	WRK_B_MAXPARM(R10)	;	
CE AA	94	00A2	230	CLRB	WRK_B_PARMCNT(R10)	;	
CF AA	94	00A5	231	CLRB	WRK_B_PARMSUM(R10)	;	
68 AB	24	AA	00A8	232	BICW	#PRC_A_DISABL:-	
00AF CB	44	BF	8A	00AC	233	PRC_A_IND, PRC_W_FLAGS(R11)	
			00B2	234	BICB	#PRC_A_RUNDEF:-	
			00B2	235		PRC_M_FLUSH, PRC_B_FLAGS2(R11)	
			00B2	236		;	
			00B2	237		;	
			00B2	238		;	
			00B2	239		;	
C5 AA	94	00B2	240	CLRB	WRK_B_RECALLCNT(R10)	;	
012F CB	00	00B5	241	MOVL	PRC_L_RECALLPTR(R11),-	;	
EA AA	00B9	242			WRK_L_RECALLPTR(R10)	;	
		00BB	243			;	
		00BB	244			;	
		00BB	245			;	
		00BB	246			;	
		00BB	247			;	
		00BB	248			;	
		00BB	249			;	
		00BB	250			;	
		00BB	251			;	
		00BB	252			;	
20 68 AB	01	E5	00BB	BBCC	#PRC_V_CNTRLY, PRC_W_FLAGS(R11), 11\$	:BRANCH IF NOT CONTROL Y/C	
FF3D'	30	00C0	253	BSBW	DCLSLOCKED STATE	:RESTORE LOCKED KEYPAD STATE	
51 F896 CA	94	00C3	254	CLRB	WRK_G_INPBUF(R10)	:ABORT TWO-PART COMMAND IN PROGRESS	
00B8 CB	00	00C7	255	MOVL	PRC_L_ONCTLY(R11), R1	:GET ADDRESS OF ACTION COMMAND	
OB	13	00CC	256	BEQL	10\$	:BR IF NO ACTION OUTSTANDING	
50 50	81	9A	00CE	MOVZBL	(R1)+ R0	:GET THE ADDRESS AND COUNT	
F896 CA	61	50	00D1	MOVC	R0, (R1), WRK_G_INPBUF(R10)	:MOVE COMMAND INTO BUFFER	
2F	11	00D7	257	BRB	15\$	:CLEAR GOTO, THEN PROCESS COMMAND	
		00D9	258	10\$:	SETBIT	:SET CONTROL Y/C LEVEL	
		00DE	259		BRB	20\$	
		00E0	260				
		00E0	261				
		00E0	262				
		00E0	263				
		00E0	264				
		00E0	265				
		00E0	266				
		00E0	267				
		00E0	268	11\$:	BSBW	DCLSLOCKED STATE	
56 00AF	CB	7E	00E3	MOVAQ	PRC_Q_COMMAND(R11), R6	:RESTORE LOCKED KEYPAD STATE	
00AF CB	00	E4	00E8	BBSC	#PRC_V_CMD, PRC_B_FLAGS2(R11), 13\$	:GET DESCRIPTOR OF COMMAND LINE	
23 00AF CB	01	E5	00EE	BBCC	#PRC_V_CHAIN, PRC_B_FLAGS2(R11), 20\$	:BRANCH IF USER COMMAND SETUP	
56 00D8 CB	7E	00F4	271	MOVAQ	PRC_Q_IMAGENAME(R11), R6	:BRANCH IF NOT USER CHAIN	
F896 CA	04	B6	00F9	272	SETBIT	PRC_V_RUNDEF, PRC_B_FLAGS2(R11)	:GET DESCRIPTOR OF IMAGE TO INVOKE
FEF8'	28	00FE	273	MOVC	(R6), 34(R6), WRK_G_INPBUF(R10)	:SET RUN DEFAULT	
		0105	274	13\$:	BSBW	DCLSALLDEACMD	:SET STRING IN BUFFER
		0108	275			:GO DEALLOCATE THE SPACE	
		0108	276				
		0108	277				
		0108	278				
		0108	279				
		0108	280				
		0108	281	15\$:	MOVAB	: IF CONTROL IS TO BE TRANSFERED TO AN ON-CONTROLY CONDITION OR A CHAINED	
		010C	282		WRK_G_INPBUF-1(R10), -	COMMAND, THEN TERMINATE ANY FORWARD GOTO IN PROGRESS.	
		010F	283	BBCC	WRK_L_CHARPTR(R10)		
03 68 AB	04	E5	284		#PRC_V_GOTO, PRC_W_FLAGS(R11), 20\$	:SET POINTER BEFORE RECORD	
F48E CA						:CLEAR GOTO FLAG	

FEE9' 30 0114 284 BSBW DCLSDEALGOTO :DEALLOCATE GOTO SYMBOL  
 0117 285  
 0117 286  
 0117 287 : IF A HANGUP PENDING, THEN ABORT THE PROCESS.  
 0117 288  
 03 68 AB 0C E1 0117 289 20\$: BBC #PRC\_V\_HANGUP,PRC\_W\_FLAGS(R11),22\$ ;IF SET, HANGUP PENDING  
 03E5 31 011C 290 BRW DCLSABORT ; LOG THE PROCESS OUT  
 011F 291  
 011F 292  
 011F 293 : IF A CHAINED IMAGE WAS SPECIFIED, THEN PROCESS THE IMAGE IMMEDIATELY AND  
 011F 294 : AS A FOREIGN COMMAND.  
 011F 295  
 06 00AF CB 02 E1 011F 296 22\$: BBC #PRC\_V\_RUNDEF,PRC\_B\_FLAGS2(R11),30\$ ;BRANCH IF NOT CHAIN IMAGE  
 FED8' 30 0125 297 BSBW DCLSMARK ;MARK CURRENT PARSE POSITION  
 012A 31 0128 298 BRW FORVERB ;PARSE AS FOREIGN COMMAND  
 012B 299  
 012B 300  
 012B 301 : BEGIN PARSING INPUT  
 012B 302  
 012B 303 : IF WE DON'T HAVE A COMMAND LINE YET, THEN FORCE AN INPUT TO HAPPEN.  
 012B 304 : LOOP TO RESTART IF THE INPUT IS NULL. STRIP OFF THE LEADING '\$'  
 012B 305 : IF IT IS PRESENT.  
 012B 306  
 FED2' 30 012B 307 30\$: BSBW DCLSSETNBLK :POINT TO NEXT NONBLANK CHARACTER  
 4E 13 012E 308 BEQL 50\$ :IF EQL NULL LINE  
 50 24 91 0130 309 35\$: CMPB #^A/\$,,R0 :DOLLAR SIGN?  
 18 12 0133 310 BNEQ 40\$ :IF NEQ NO-PROCESS THE COMMAND  
 F486 CA F492 CA 9E 0135 311 MOVAB WRK\_G\_BUFFER(R10),WRK\_L\_EXPANDPTR(R10) ;RESET EXPANSION BUFFER POINT  
 FEC1' 30 013C 312 BSBW DCLSMOVCHAR ;PUT DOLLAR SIGN IN BUFFER  
 7C AB DD 013F 313 PUSHL PRC\_L\_INDCLOCK(R11) ;SAVE CURRENT INDIRECT CHANGE COUNT  
 FEBB' 30 0142 314 BSBW DCLSSETNBLK ;POINT AT NEXT NON-BLANK CHARACTER  
 37 13 0145 315 BEQL 50\$ ;BRANCH IF NULL LINE TO DCLSRESTART  
 8E 7C AB D1 0147 316 CMPL PRC\_L\_INDCLOCK(R11),(SP)+ ;INDIRECT LEVEL CHANGE?  
 E3 12 014B 317 BNEQ 35\$ ;IF YES-LOOK FOR LEADING DOLLAR SIGN  
 014D 318  
 014D 319 : GET THE FIRST TOKEN FROM THE COMMAND LINE. IT SHOULD BE EITHER A VERB,  
 014D 320 : A SYMBOL, OR A LABEL.  
 014D 321  
 014D 322  
 FEB0' 30 014D 323 40\$: BSBW DCLSMARK ;MARK CURRENT PARSE POSITION  
 FEAD' 30 0150 324 BSBW DCLSGETOKEN ;GET COMMAND VERB, SYMBOL, OR LABEL  
 03 12 0153 325 BNEQ 41\$ ;IF EQL NO TOKEN WAS FOUND  
 0123 31 0155 326 BRW NOCOMD ;SIGNAL BAD COMMAND LINE SYNTAX  
 0158 327  
 0158 328 : CHECK FOR COLON TERMINATOR. IT COULD INDICATE EITHER A LABEL OR A NULL  
 0158 329 : NODE.  
 0158 330  
 0158 331  
 50 7E D4 0158 332 41\$: CLRL -(SP) :ASSUME TERMINATOR IS NOT A BLANK  
 20 91 015A 333 CMPB #^A' ',R0 :IS TERMINATOR A BLANK?  
 0D 12 015D 334 BNEQ 42\$ :NO, THEN BRANCH  
 6E D6 015F 335 INCL (SP) :SET BLANK SEEN FLAG  
 FE98' 30 0161 336 SETBIT PRC\_V\_IND,PRC\_W\_FLAGS(R11) :TEMPORARILY DISALLOW INDIRECT  
 0165 337 BSBW DCLSSETNBLK :PEEK AT NEXT NON-BLANK  
 50 3A 91 016C 338 CLRBIT PRC\_V\_IND,PRC\_W\_FLAGS(R11) :RESTORE INDIRECT  
 10 13 016F 339 42\$: CMPB #^A':',R0 :LABEL OR STRING EQUATE TERMINATOR?  
 BEQL 60\$ :BRANCH IF SO

50 8E D5 0171 341 TSTL (SP)+  
06 13 0173 342 BEQL 438 : WAS BLANK SEEN?  
20 90 0175 343 MOVB #^A^, R0 : BRANCH IF NO BLANK SEEN  
FE85' 30 0178 344 BSBW DCLSBACKUPCHAR : RESTORE THE BLANK  
005D 31 017B 345 43\$: BRW SYMBOL : ELSE, VERB OR ASSIGNMENT STATEMENT

017E 346 :  
017E 347 : IF NULL LINE, THEN GET NEXT COMMAND.  
FE7F 31 017E 348 50\$: BRW DCLSRESTART : GET NEXT COMMAND  
01B1 349 :  
01B1 350 :  
01B1 351 :  
01B1 352 :  
01B1 353 : LABEL OR STRING ASSIGNMENT  
01B1 354 :  
01B1 355 : THE FIRST TOKEN WAS TERMINATED BY A ":". IF IT IS NOT IMMEDIATELY FOLLOWED  
01B1 356 : BY A SECOND COLON, I.E., IS A NULL NODE SPEC, IT MUST BE EITHER A  
01B1 357 : LABEL OR THE SYMBOL NAME IN A STRING ASSIGNMENT. WE TEMPORARILY DISALLOW  
01B1 358 : INDIRECT WHEN PEEKING AT THE NEXT CHARACTER AFTER THE COLON, SO THAT IF  
01B1 359 : AN "A" APPEARS JUST AFTER THE LABEL, WE DON'T GET SHIFTED TO THE NEXT LEVEL  
01B1 360 : BEFORE WE GET A CHANCE TO PUT THE LABEL ENTRY IN THE RIGHT TABLE.

FE7C' 30 01B1 361 60\$: BSBW DCLSMOVCHAR : MOVE CHARACTER TO COMMAND BUFFER  
01B4 362 SETBIT PRC\_V\_IND, PRC\_W\_FLAGS(R11) : TEMPORARILY DISALLOW INDIRECT  
FE75' 30 01B8 363 BSBW DCLSSETCHAR : PEEK AT NEXT CHARACTER IN INPUT BUF  
50 3A 91 01B8 364 CLRBIT PRC\_V\_IND, PRC\_W\_FLAGS(R11) : RESTORE INDIRECT  
3A 13 0192 365 CMPB #^A7:7, R0 : COLON?  
8E D5 0194 366 BEQL NULLNODE : IF EQL YES  
50 3D 91 0196 367 TSTL (SP)+ : RESTORE THE STACK  
2E 13 0199 368 CMPB #^A/=, R0 : EQUAL SIGN?  
019B 369 BEQL 80\$ : IF EQL YES

019B 370 :  
019B 371 :  
019B 372 :  
019B 373 :  
019B 374 :  
019B 375 : LABEL  
019B 376 : SAVE THE LABEL IN THE SYMBOL TABLE. IF THE A LABEL OF THE SAME NAME WAS  
019B 377 : ALREADY FOUND, REPLACE IT WITH THE ONE WE HAVE JUST FOUND.

1A 68 AB 0B E0 019B 378 BBS #PRC\_V\_YLEVEL- : IF SET, AT CONTROL Y/C LEVEL  
50 14 AB D0 019D 379 MOVL PRC\_Q\_FLAGS(R11), 69\$ : DEVICE CHAR SAVED IN CURRENT RAB  
14 18 A0 02 E0 01A0 380 BBS PRC\_L\_INDINPRAB(R11), R0 : IF TRM, ISSUE WARNING MESSAGE  
OC 18 A0 1C E1 01A4 381 BBC #DEV\$0\_TRM, RABSL\_CTX(R0), 65\$ : IF NOT RANDOM DEVICE, IGNORE IT  
FE49' 30 01B4 382 DISABLE #DEV\$V\_RND, RABSL\_CTX(R0), 69\$ : DISABLE CONTROL Y/C AST'S  
34 50 F9 01B7 383 BSBW DCLSALLOC\_LABEL : ALLOCATE AND INSERT LABEL IN TABLE  
FE54 31 01BA 384 BLBC R0, 130\$ : IF LBC ALLOCATION FAILURE  
01BD 385 69\$: BRW DCLS\_CMDSTART : START COMMAND OVER  
F1 11 01C7 386 65\$: ERMMSG NOLBLS : SIGNAL LABEL NOT ALLOWED HERE  
01C9 387 BRB 69\$ :  
01C9 388 :  
01C9 389 :  
01C9 390 :  
01C9 391 :  
01C9 392 :  
01C9 393 : STRING ASSIGNMENT  
01C9 394 : WE HAVE DETECTED A ":"=:" FOLLOWING THE FIRST TOKEN ON THE LINE.  
01C9 395 : THEREFORE WE MUST BE PROCESSING A STRING ASSIGNMENT.

FE34' 30 01C9 396 80\$: BSBW DCLS\_EQUATE : PROCESS STRING EQUATE STATEMENT  
20 11 01CC 397 BRB 130\$ :  
:

01CE 398  
 01CE 399  
 01CE 400 : NULL NODE SPECIFICATION WAS FOUND. BACK UP PAST DOUBLE COLONS AND PROCESS  
 01CE 401 : AS A VERB.  
 01CE 402  
 01CE 403 : NULLNODE:  
 FE2F' 30 01CE 404 BSBW DCLSBACKUPMOVE : BACK UP TO END OF VERB NAME  
 8E D5 01D1 405 TSTL (SP)+ : WAS BLANK SEEN?  
 50 06 13 01D3 406 BEQL SYMBOL : BRANCH IF NO BLANK SEEN  
 20 90 01D5 407 MOVB #^A' R0 : RESTORE THE BLANK  
 FE25' 30 01DB 408 BSBW DCLSBACKUPCHAR :  
 01DB 409  
 01DB 410  
 01DB 411 : COMMAND VERB, INTEGER ASSIGNMENT, SUBSTRING ASSIGNMENT, OR BITFIELD ASSIGNMENT  
 01DB 412  
 01DB 413 : THE FIRST TOKEN WAS NOT TERMINATED BY A ":", IT MUST BE EITHER A COMMAND VERB  
 01DB 414 : OR THE SYMBOL NAME IN AN INTEGER, SUBSTRING, OR BITFIELD ASSIGNMENT.  
 01DB 415  
 01DB 416 : PROCESS INTEGER, SUBSTRING, OR BITFIELD ASSIGNMENT  
 01DB 417  
 50 5B 8F 91 01DB 418 SYMBOL: CMPB #^A/E/,R0 : SUBSTRING OF BIT FIELD SUBSTRING ASSIGNMENT  
 05 12 01DF 419 BNEQ 120\$ : IF NEQ NO  
 FE1C' 30 01E1 420 BSBW DCLSASSIGN : PROCESS SUBSTRING OR BIT FIELD ASSIGNMENT  
 08 11 01E4 421 BRB 130\$ :  
 50 3D 91 01E6 422 120\$: CMPB #^A/=/,R0 : SYMBOL ASSIGNMENT?  
 06 12 01E9 423 BNEQ COMMAND : IF NEQ NO  
 FE12' 30 01EB 424 BSBW DCLSEVALUATE : EVALUATE ARITHMETIC ASSIGNMENT STATEMENT  
 0146 31 01EE 425 130\$: BRU ERROR\_EXIT :  
 01F1 426  
 01F1 427 : COMMAND VERB  
 01F1 428  
 01F1 429  
 01F1 430 : WE HAVE DETERMINED THAT WE ARE PARSING A COMMAND THAT BEGINS WITH A VERB.  
 01F1 431 : TRANSLATE THE FIRST TOKEN ON THE COMMAND LINE (IF POSSIBLE).  
 01F1 432  
 06 88 01F1 433 : COMMAND: PUSHR #^M<R1,R2> : SAVE COMMAND VERB PARAMETERS  
 54 FEOA' 30 01F3 434 BSBW DCLSSTRING : SEARCH FOR SYMBOL DEFINITION  
 51 7D 01F6 435 MOVQ R1,R4 : SAVE VALUE PARAMETERS  
 06 BA 01F9 436 POPR #^M<R1,R2> : RESTORE COMMAND VERB PARAMETERS  
 54 D5 01FB 437 TSTL R4 : SYMBOL DEFINED AND HAVE VALUE?  
 03 12 01FD 438 BNEQ 135\$ : IF NEQ YES  
 0083 31 01FF 439 BRU PRCVERB : IF NOT SYMBOL, SEARCH VERB TABLE  
 0202 440  
 0203 441  
 0203 442 : THE FIRST TOKEN IS A SYMBOL. GET ITS VALUE.  
 0203 443  
 F486 CA F492 CA 9E 0202 444 135\$: MOVAB WRK\_G BUFFER(R10),WRK\_L EXPANDPTR(R10) : RESET EXPANSION BUFFER POINT  
 56 F48E CA 01 C1 0209 445 ADDL3 #1,WRK\_L CHARPTR(R10),R6 : GET CHARACTER POINTER  
 66 20 91 020F 446 CMPB #^A/ /,(R6) : ALREADY A BLANK HERE?  
 03 13 0212 447 BEQL 140\$ : IF EQL YES  
 76 20 90 0214 448 MOVAB #^A/ /,-(R6) : ELSE INSERT A TRAILING BLANK  
 FF A544 27 91 0217 449 140\$: CMPB #^A/ /,-1(R5)[R4] : DOES THE SYMBOL END WITH '?  
 04 12 021C 450 BNEQ 150\$ : IF NEQ NO  
 54 D7 021E 451 DECL R4 : ELSE REMOVE THE ' FROM THE SYMBOL  
 56 D6 0220 452 INCL R6 : AND REMOVE THE ADDED TRAILING BLANK  
 56 54 C2 0222 453 150\$: SUBL R4,R6 : BACK UP OVER SYMBOL'S LENGTH  
 F48E CA 56 01 C3 0225 454 SUBL3 #1,R6,WRK\_L\_CHARPTR(R10) : STORE NEW CHARACTER POINTER

66 65 54 28 022B 455 MOVC R4, (R5), (R6) :MOVE TRANSLATED COMMAND TO INPUT BUFFER  
 FDCA' 30 022F 456 SETBIT PRC\_V\_IND, PRC\_W\_FLAGS(R11) :TEMPORARILY DISALLOW INDIRECT  
     30 0233 457 BSBW DCL\$SETNBLK :SKIP OVER LEADING BLANKS  
     50 03 0236 458 CLRBIT PRC\_V\_IND, PRC\_W\_FLAGS(R11) :ENABLE INDIRECT  
     03 12 023A 459 TSTB R0 :COMMAND NOP'D?  
 50 40 8F 91 0241 460 BNEQ 152\$ :BR IF NOT  
     03 12 0245 461 BRW DCLSRESTART :  
 FDBF 31 023E 462 152\$: CMPB #^A/8/, R0 :PROCEDURE FILE?  
     FDC7 31 0247 463 BNEQ 155\$ :IF NEQ NO  
     31 024A 464 BRW DCLSCHDSTART :READ FROM THE COMMAND FILE  
     024A 465 :  
     024A 466 :  
     024A 467 : IF THE SYMBOL DEFINES A FOREIGN COMMAND, THEN PROCESS IT AS SUCH.  
     , 024A 468 : OTHERWISE, SEARCH THE VERB TABLE FOR THE VERB.  
     024A 469 :  
 50 FDB3' 30 024A 470 155\$: BSBW DCLSMARK :MARK CURRENT PARSE POSITION  
 FDB0' 30 024D 471 BSBW DCLSSETNBLK :POINT TO NEXT NONBLANK CHARACTER  
     24 91 0250 472 CMPB #^A/\$/, R0 :DOLLAR SIGN?  
     21 12 0253 473 BNEQ CMDVERB :IF NEQ NO  
     0255 474 :  
     0255 475 DSABL LSB :  
     0255 476 :  
     0255 477 :  
     0255 478 FOREIGN COMMAND :  
     0255 479 :  
     0255 480 : PROCESS THE FILE SPECIFICATION AND THEN PROCESS THE REST OF THE COMMAND  
     0255 481 : LINE.  
     0255 482 :  
     0255 483 FORVERB: :  
 53 03 9A 0255 484 MOVZBL #PTR K PARAMETR, R3 :COME HERE FOR FOREIGN VERBS  
 FDAS' 30 0258 485 BSBW DCLSPROCFILE :SET TOKEN CONTEXT OF FILESPEC  
     24 50 E9 025B 486 BLBC R0\_ERROR :PROCESS FILE SPECIFICATION  
     BA AA 0C C2 025E 487 SUBL #PTR C LENGTH, WRK\_L\_RSLNXT(R10) :IF LBC FILE SPECIFICATION ERROR  
     7E 51 7D 0262 488 MOVO R1, -TSP) :UNDO RESULT TOKEN DESCRIPTOR  
     0347 30 0265 489 BSBW PROCFORN :SAVE FILENAME DESCRIPTOR PARAMETERS  
 C2 AA 00'8F 90 0268 490 MOVB #CLISK VERB\_FORE, WRK\_B\_VERBTYP(R10) :PROCESS FOREIGN COMMAND  
     06 BA 026D 491 POPR #^M<R1, R2> :SET FOREIGN COMMAND  
 53 0E'8F 9A 026F 492 MOVZBL #IMG K EXTIMAGE<^X7F, R3 :RESTORE FILENAME DESCRIPTOR PARAMETERS  
     007A 31 0273 493 BRW EXECEXT :SET EXTERNAL IMAGE INDEX  
     0276 494 :  
     0276 495 : GET THE TRANSLATED COMMAND VERB :  
     0276 496 :  
 FD87' 30 0276 497 CMDVERB: :  
     0A 12 0279 499 BSBW DCLSGETOKEN :GET COMMAND VERB TOKEN  
     00B2 31 027B 500 BNEQ PRCVERB :PROCEED IF VALID  
     0282 501 NOCMD: STATUS NOCMD :ERROR IF NULL LINE  
     0282 502 ERROR: BRW ERROR\_EXIT :  
     0285 503 :  
     0285 504 : LOOK THE VERB UP IN THE COMMAND TABLES.  
     0285 505 :  
     0285 506 :  
     0285 507 PREVERB: :  
 09 68 AB 08 E1 0285 508 BBC #PRC\_V\_YLEVEL, PRC\_W\_FLAGS(R11), 10\$ :BR IF NOT AT "Y" LEVEL  
 04 00 BB 18 E1 028A 509 BBC #PSL\$V\_CURMOD, #PRC\_C\_SAVAP(R11), 10\$ :IF CLR PREVIOUS MODE SUPER  
 58 00000000'GF DD 0293 511 10\$: SETBIT PRC\_V\_IND, PRC\_W\_FLAGS(R11) :DISABLE "A" FILE RECOGNITION  
     028F 510 MOVL G\$CTL\$AG\_CLITABCE, R8 :GET ADDRESS OF DATA BASE VECTOR

FD63' 30 029A 512      BSBW      DCLSSEARCH\_VERB      ;SEARCH VERB TABLE FOR VERB  
 E2 50 E9 029D 513      BLBC      RO,ERROR      ;BRANCH IF ERROR

02A0 514  
 02A0 515  
 02A0 516  
 02A0 517  
 02A0 518  
 02A0 519  
 02A0 520  
 02A0 521  
 02A0 522  
 02A0 523      ; IF THE COMMAND IS DEFINED IN THE TABLES AS FOREIGN THEN PERFORM THE SPECIAL  
 02A0 524      ; COMMAND LINE PROCESSING.  
 02A0 525  
 02A0 526      ; IF IT IS DEFINED AS IMMEDIATE MODE AND THEREFORE, SHOULD BE DISPATCHED  
 02A0 527      ; IMMEDIATELY, DO IT NOW.  
 02A0 528  
 02A0 529      ; OTHERWISE, PARSE THE COMMAND PARAMETERS AND QUALIFIERS.  
 02A0 530      ;  
 02A0 531      ;  
 02A0 532      ;  
 02A0 533      ;  
 02A0 534      ;  
 02A0 535      ;  
 02A0 536      ;  
 02A0 537      ;  
 02A0 538      ;  
 02A0 539      ;  
 02A0 540      ;  
 02A0 541      ;  
 02A0 542      ;  
 02A0 543      ; BUILD THE IMAGE/ROUTINE NAME DESCRIPTOR  
 02A0 544      ;  
 02A0 545      ; BUILD\_IMAGE:  
 02A0 546      ;  
 02A0 547      ;  
 02A0 548      ;  
 02A0 549      ;  
 02A0 550      ;  
 02A0 551      ;  
 02A0 552      ;  
 02A0 553      ;  
 02A0 554      ;  
 02A0 555      ;  
 02A0 556      ;  
 02A0 557      ;  
 02A0 558      ;  
 02A0 559      ;  
 02A0 560      ;  
 02A0 561      ;  
 02A0 562      ;  
 02A0 563      ;  
 02A0 564      ;  
 02A0 565      ;  
 02A0 566      ;  
 02A0 567      ;  
 02A0 568      ;  
 02A0 524      BBC      #CMD V FOREIGN -  
 02A0 525      CMD @ FLAGS(R8),20\$      ;BRANCH IF NOT FOREIGN  
 02A0 526      BSBW      PROCFORN      ;PROCESS FOREIGN COMMAND  
 02A0 527      BRB      BUILD\_IMAGE      ;PROCESS THE IMAGE NAME  
 02A0 528      20\$: BBC      #CMD V IMMED,CMD W FLAGS(R8),30\$      ;BRANCH IF NOT IMMEDIATE  
 02A0 529      MOVL      WRK C IMAGE(R10),R2      ;GET ADDR OF ASCIC ROUTINE NAME  
 02A0 530      MOVZBL      (R2)+,R1      ;GET COUNT IN R1, ADDRESS IN R2  
 02A0 531      BSBW      DCLSLOCATE\_INTERNAL      ;LOCATE INTERNAL ROUTINE INDEX  
 02A0 532      BLBC      RO,ERROR      ;BRANCH IF ERROR  
 02A0 533      BRB      IMMED      ;DISPATCH THE IMMEDIATE COMMAND  
 02A0 534      30\$: BSBW      DCLSPARSE COMMAND      ;PARSE THE COMMAND QUALS AND PARMs  
 02A0 535      CMPL      RO #CLIS\_NOCMD      ;IF CTRL/Z WAS ENTERED,  
 02A0 536      BNEQ      40\$      ;THEN GET A NEW COMMAND  
 02A0 537      BRW      DCLSRESTART      ;  
 02A0 538      BLBC      RO,ERROR      ;SIGNAL ANY SYNTAX ERRORS  
 02A0 539      40\$:      ;  
 02A0 540      ;  
 02A0 541      ;  
 02A0 542      ;  
 02A0 543      ;  
 02A0 544      ;  
 02A0 545      ;  
 02A0 546      MOVL      WRK L IMAGE(R10),R2      ;GET IMAGE/ROUTINE ASCIC NAME ADDRES  
 02A0 547      MOVZBL      (R2)+,R1      ;GET COUNT IN R1, ADDRESS IN R2  
 02A0 548      MOVZBL      #IMG\_K\_EXTIMAGE@X7F,R3      ;ASSUME EXTERNAL IMAGE  
 02A0 549      BBS      #WRK V CLIRTN -      ;BRANCH IF INTERNAL ROUTINE  
 02A0 550      WRK @ FLAGS2(R10),5\$      ;  
 02A0 551      BRB      EXECEXT      ;EXECUTE EXTERNAL COMMAND  
 02A0 552      BSBW      DCLSLOCATE\_INTERNAL      ;LOCATE INTERNAL ROUTINE INDEX  
 02A0 553      BLBC      RO,ERROR      ;BRANCH IF ERROR  
 02A0 554      55\$:      ;  
 02A0 555      ;  
 02A0 556      ;  
 02A0 557      ;  
 02A0 558      ;  
 02A0 559      PUSHL      R3      ;SAVE COMMAND INDEX  
 02A0 560      BSBW      DCLSSORT\_TOKENS      ;SORT DESCRIPTOR TABLE  
 02A0 561      POPL      R3      ;RESTORE COMMAND INDEX  
 02A0 562      56\$: EXECEXT:      ;  
 02A0 563      SETBIT      WRK\_V\_COMMAND,WRK\_W\_FLAGS(R10)      ;SET COMMAND EXECUTION IN PROGRESS  
 02A0 564      MOVAB      WRK\_G\_RESULT(R10),WRK\_L\_RSLNXT(R10)      ;RESET NEXT TOKEN TO RETRIEVE  
 02A0 565      567:      ;  
 02A0 566      ;  
 02A0 567      ;  
 02A0 568      ; INTERNAL COMMAND PARAMETERS:

02FA 569 :  
 02FA 570 :  
 02FA 571 :  
 02FA 572 :  
 02FA 573 :  
 02FA 574 :  
 02FA 575 :  
 02FA 576 :  
 02FA 577 :  
 02FA 578 :  
 02FA 579 :  
 02FA 580 :  
 02FA 581 :  
 02FA 582 :  
 02FA 583 :  
 02FA 584 :  
 02FA 585 :  
 02FA 586 :  
 02FA 587 :  
 0300 588 :  
 0300 589 :  
 0300 590 :  
 0300 591 :  
 53 03'8F 91 0300 592 :  
 1D 13 0304 593 :  
 53 28'8F 91 0306 594 :  
 17 13 030A 595 :  
 53 29'8F 91 030C 596 :  
 11 13 0310 597 :  
 05 F0 AA 01 E3 0312 598 :  
 FCE6' 30 0317 599 :  
 07 11 031A 600 :  
 FCE1' 30 031C 601 :  
 031F 602 :  
 0323 603 :  
 0323 604 :  
 0323 605 :  
 0323 606 :  
 0323 607 :  
 5E FE00 5E D0 0323 608 :  
 CE 9E 0326 609 :  
 7E 0200 8F 3C 0328 610 :  
 58 5E D0 032D 611 :  
 12 10 0332 612 :  
 0335 613 :  
 0337 614 :  
 0337 615 :  
 0337 616 :  
 0337 617 :  
 0337 618 :  
 0337 619 :  
 03 50 E8 0337 620 :  
 FCC3' 30 033A 621 :  
 00000000'FF 16 033D 622 :  
 023A 30 0343 623 :  
 FCB7 31 0346 624 :  
 0349 625 :  
 - PROCESS NEXT COMMAND  
PROCESS REST OF COMMAND

R1 = LENGTH OF IMAGE FILENAME (IF ANY).  
 R2 = ADDRESS OF IMAGE FILENAME (IF ANY).  
 R8 AND R9 ARE SET UP AFTER THE CONDITIONAL RUNDOWN CALL BELOW  
 R8 = ADDRESS OF SCRATCH BUFFER DESCRIPTOR.  
 R9 = ADDRESS OF SCRATCH STACK.  
 R10 = BASE ADDRESS OF COMMAND WORK AREA.  
 R11 = BASE ADDRESS OF PROCESS WORK AREA.

IMMEDIATE COMMAND PARAMETERS:  
 R10 = BASE ADDRESS OF COMMAND WORK AREA.  
 R11 = BASE ADDRESS OF PROCESS WORK AREA.

IMMED: BBC #PRC\_V\_PRIV,PRC\_B\_FLAGS2(R11),10\$ ;IMMEDIATE COMMAND EXECUTION  
 NEED TO RUN DOWN PRIVILEGED IMAGE, BUT ALLOW THE FOLLOWING COMMANDS  
 CMPB #IMG\_K\_CONTINUE & "X7F,R3 :CONTINUE COMMAND?  
 BEQL 10\$ ;BR IF YES  
 CMPB #IMG\_K\_SPAWN & "X7F,R3 ;SPAWN COMMAND?  
 BEQL 10\$ ;BR IF YES  
 CMPB #IMG\_K\_ATTACH & "X7F,R3 ;ATTACH COMMAND?  
 BEQL 10\$ ;BR IF YES  
 BBCS #WRK\_V\_COMMAND,WRK\_W\_FLAGS(R10),2\$  
 BSBW DCLSFORCEEXIT ;RUN DOWN THE PRIVILEGED IMAGE  
 BRB 10\$ ;  
 2\$: BSBW DCLSFORCEEXIT ;RUN DOWN THE PRIVILEGED IMAGE  
 CLRBIT WRK\_V\_COMMAND,WRK\_W\_FLAGS(R10)  
 : SETUP SCRATCH STORAGE FOR USE BY INTERNAL ROUTINES  
 10\$: MOVL SP,R9 ;SET ADDRESS OF SCRATCH STACK  
 MOVAB -WRK\_C\_SCRSTKSIZ(SP),SP ;ALLOCATE SCRATCH AREA BEFORE STACK  
 PUSHAB (SP) ;BUILD SCRATCH BUFFER DESCRIPTOR  
 MOVZWL #WRK\_C\_SCRSTKSIZ,-(SP)  
 MOVL SP,R8 ;SET ADDRESS OF SCRATCH BUFFER DESCRIPTOR  
 BSBB IMAGECASE ;EXECUTE INTERNAL IMAGE

THE FOLLOWING CODE IS DUPLICATED IN IMAGEEXEC FOR EXTERNAL IMAGES  
 BECAUSE IT MUST BE DONE BEFORE IMAGE RUNDOWN, NOT AFTERWARDS.

ERROR\_EXIT:  
 BLBS R0,10\$ ;ERROR EXIT  
 BSBW DCLSERRMSG ;IF LBS SUCCESSFUL COMPLETION  
 10\$: JSB DCLSSET\_STATUS ;OUTPUT SYSTEM ERROR MESSAGE  
 BSBW DCLSFLUSH ;SET COMPLETION STATUS  
 BRW DCLSRESTART ;FLUSH COMMAND BUFFER  
 ;

0349 627 .MACRO INTIMAGE NAME  
 0349 628 .WORD XXX\$'NAME'-10\$  
 0349 629 IMG\_K 'NAME = \$INTIMAGES  
 0349 630 \$INTIMAGES = \$INTIMAGES + 1  
 0349 631 .ENDM  
 0349 632  
 35'8F 00 53 8F 0349 633 IMAGECASE:  
 034E 634 CASEB R3,#0,#<90\$-10\$>/2-1  
 034E 635 10\$: ;BASE ADDRESS OF CASE TABLE  
 034E 636 ;GENERATE INTERNAL IMAGE CASE TABLE  
 034E 637 LIST MEB  
 0072' 034E INTIMAGES  
 0078' 0350 .WORD XXX\$ALLOCATE-10\$  
 007E' 0352 .WORD XXX\$ASSIGN-10\$  
 0084' 0354 .WORD XXX\$CLOSE-10\$  
 008A' 0356 .WORD XXX\$CONTINUE-10\$  
 0090' 0358 .WORD XXX\$DEALLOCAT-10\$  
 0096' 035A .WORD XXX\$DEASSIGN-10\$  
 009C' 035C .WORD XXX\$DEBUG-10\$  
 00A2' 035E .WORD XXX\$DECK-10\$  
 00A8' 0360 .WORD XXX\$DEFINE-10\$  
 00AE' 0362 .WORD XXX\$DEPOSIT-10\$  
 00B4' 0364 .WORD XXX\$DELSYM-10\$  
 00BA' 0366 .WORD XXX\$EOOD-10\$  
 00C0' 0368 .WORD XXX\$EXAMINE-10\$  
 00C6' 036A .WORD XXX\$EXIT-10\$  
 00CC' 036C .WORD XXX\$EXTIMAGE-10\$  
 00D2' 036E .WORD XXX\$GOTO-10\$  
 00D8' 0370 .WORD XXX\$IF-10\$  
 00DE' 0372 .WORD XXX\$INQUIRE-10\$  
 00E4' 0374 .WORD XXX\$LOGOUT-10\$  
 00EA' 0376 .WORD XXX\$MCR-10\$  
 00F0' 0378 .WORD XXX\$ON-10\$  
 00F6' 037A .WORD XXX\$OPEN-10\$  
 00FC' 037C .WORD XXX\$READ-10\$  
 0102' 037E .WORD XXX\$RUN-10\$  
 0108' 0380 .WORD XXX\$SETCTLY-10\$  
 010E' 0382 .WORD XXX\$SETDEFAULT-10\$  
 0114' 0384 .WORD XXX\$SETTON-10\$  
 011A' 0386 .WORD XXX\$SETUIC-10\$  
 0120' 0388 .WORD XXX\$SETVERIFY-10\$  
 0126' 038A .WORD XXX\$SHOWDEF-10\$  
 012C' 038C .WORD XXX\$SHOWPROT-10\$  
 0132' 038E .WORD XXX\$SHOWQUOTA-10\$  
 0138' 0390 .WORD XXX\$SHOWSTAT-10\$  
 013E' 0392 .WORD XXX\$SHOWSYMBL-10\$  
 0144' 0394 .WORD XXX\$SHOWTIME-10\$  
 014A' 0396 .WORD XXX\$SHOWTRAN-10\$  
 0150' 0398 .WORD XXX\$STOP-10\$  
 0156' 039A .WORD XXX\$WAIT-10\$  
 015C' 039C .WORD XXX\$WRITE-10\$  
 0162' 039E .WORD XXX\$SPAWN-10\$  
 0168' 03A0 .WORD XXX\$ATTACH-10\$  
 016E' 03A2 .WORD XXX\$EOJ-10\$  
 0174' 03A4 .WORD XXX\$CANCEL-10\$  
 017A' 03A6 .WORD XXX\$SETPROMPT-10\$  
 0180' 03A8 .WORD XXX\$RECALL-10\$

0186' 03AA	.WORD	XXX\$DEFKEY-10\$
018C' 03AC	.WORD	XXX\$SHOWKEY-10\$
0192' 03AE	.WORD	XXX\$DELKEY-10\$
0198' 03B0	.WORD	XXX\$SETKEY-10\$
019E' 03B2	.WORD	XXX\$SETFLUSH-10\$
01A4' 03B4	.WORD	XXX\$CRETABLE-10\$
01AA' 03B6	.WORD	XXX\$CONNECT-10\$
01B0' 03B8	.WORD	XXX\$DISCONNECT-10\$
03BA 638	.NLIST	MEB
50 0000'8F 3C 05	03BA 640 90\$: RSB	MOVZWL #SSS_ILLSER,RO :SET SERVICE ERROR
03BF 641		
03C0 642		
03C0 643		
03C0 644	MACRO INTIMAGE NAME	
03C0 645	XXX\$'NAME': JMP DCLS'NAME'	
03C0 646	IMG K 'NAME = SINTIMAGES	
03C0 647	SINTIMAGES\$ = SINTIMAGES + 1	
03C0 648	.ENDM	
03C0 649	.LIST MEB	
03C0 650	INTIMAGES	:GENERATE INTERNAL IMAGE CASE TABLE
00000000'EF 17	03C0 XXX\$ALLOCATE: JMP DCLSALLOCATE	
00000000'EF 17	03C6 XXX\$ASSIGN: JMP DCLSASSIGN	
00000000'EF 17	03CC XXX\$CLOSE: JMP DCLSCLOSE	
00000000'EF 17	03D2 XXX\$CONTINUE: JMP DCLSCONTINUE	
00000000'EF 17	03D8 XXX\$DEALLOCAT: JMP DCLSDEALLOCAT	
00000000'EF 17	03DE XXX\$DEASSIGN: JMP DCLSDEASSIGN	
00000000'EF 17	03E4 XXX\$DEBUG: JMP DCLSDEBUG	
00000547'EF 17	03EA XXX\$DECK: JMP DCLSDECK	
00000000'EF 17	03F0 XXX\$DEFINE: JMP DCLSDEFINE	
00000000'EF 17	03F6 XXX\$DEPOSIT: JMP DCLSDEPOSIT	
00000000'EF 17	03FC XXX\$DELSYM: JMP DCLSDELSYM	
0000053A'EF 17	0402 XXX\$EOOD: JMP DCLSEOD	
00000000'EF 17	0408 XXX\$EXAMINE: JMP DCLSEXAMINE	
00000000'EF 17	040E XXX\$EXIT: JMP DCLSEXIT	
00000000'EF 17	0414 XXX\$EXTIMAGE: JMP DCLSEXIMAGE	
00000000'EF 17	041A XXX\$GOTO: JMP DCLSGOTO	
00000000'EF 17	0420 XXX\$IF: JMP DCLSIF	
00000000'EF 17	0426 XXX\$INQUIRE: JMP DCLSINQUIRE	
00000000'EF 17	042C XXX\$LOGOUT: JMP DCLSLOGOUT	
00000000'EF 17	0432 XXX\$MCR: JMP DCLSMCR	
00000000'EF 17	0438 XXX\$ON: JMP DCLSON	
00000000'EF 17	043E XXX\$OPEN: JMP DCLSOPEN	
00000000'EF 17	0444 XXX\$READ: JMP DCLSREAD	
00000000'EF 17	044A XXX\$RUN: JMP DCLSRUN	
00000000'EF 17	0450 XXX\$SETCTLY: JMP DCLSSETCTLY	
00000000'EF 17	0456 XXX\$SETDEFAULT: JMP DCLSSETDEFAULT	
00000000'EF 17	045C XXX\$SETTON: JMP DCLSSETTON	
00000000'EF 17	0462 XXX\$SETPROT: JMP DCLSSETPROT	
00000000'EF 17	0468 XXX\$SETUIC: JMP DCLSSETUIC	
00000000'EF 17	046E XXX\$SETVERIFY: JMP DCLSSETVERIFY	
00000000'EF 17	0474 XXX\$SHOWDEF: JMP DCLSSHOWDEF	
00000000'EF 17	047A XXX\$SHOWPROT: JMP DCLSSHOWPROT	
00000000'EF 17	0480 XXX\$SHOWQUOTA: JMP DCLSSHOWQUOTA	
00000000'EF 17	0486 XXX\$SHOWSTAT: JMP DCLSSHOWSTAT	
00000000'EF 17	048C XXX\$SHOWSYMBL: JMP DCLSSHOWSYMBL	
00000000'EF 17	0492 XXX\$SHOWTIME: JMP DCLSSHOWTIME	

00000000'EF	17	0498	XXX\$SHOWTRAN:	JMP	DCL\$SHOWTRAN
00000000'EF	17	049E	XXX\$STOP:	JMP	DCL\$STOP
00000000'EF	17	04A4	XXX\$WAIT:	JMP	DCL\$WAIT
00000000'EF	17	04AA	XXX\$WRITE:	JMP	DCL\$WRITE
00000000'EF	17	04B0	XXX\$SPAWN:	JMP	DCL\$SPAWN
00000000'EF	17	04B6	XXX\$ATTACH:	JMP	DCL\$ATTACH
00000504'EF	17	04Bc	XXX\$EOJ:	JMP	DCL\$EOJ
00000000'EF	17	04C2	XXX\$CANCEL:	JMP	DCL\$CANCEL
00000000'EF	17	04C8	XXX\$SETPROMPT:	JMP	DCL\$SETPROMPT
00000000'EF	17	04CE	XXX\$RECALL:	JMP	DCL\$RECALL
00000000'EF	17	04D4	XXX\$DEFKEY:	JMP	DCL\$DEFKEY
00000000'EF	17	04DA	XXX\$SHOWKEY:	JMP	DCL\$SHOWKEY
00000000'EF	17	04E0	XXX\$DELKEY:	JMP	DCL\$DELKEY
00000000'EF	17	04E6	XXX\$SETKEY:	JMP	DCL\$SETKEY
00000000'EF	17	04Ec	XXX\$SETFLUSH:	JMP	DCL\$SETFLUSH
00000000'EF	17	04F2	XXX\$CRETABLE:	JMP	DCL\$CRETABLE
00000000'EF	17	04F8	XXX\$CONNECT:	JMP	DCL\$CONNECT
00000000'EF	17	04FE	XXX\$DISCONNECT:	JMP	DCL\$DISCONNECT
		0504	.NLIST	MEB	

0504 653 .SBTTL CALL LOGINOUT TO ABORT THE PROCESS  
 0504 654 ----  
 0504 655 DCLSABORT  
 0504 656  
 0504 657  
 0504 658  
 0504 659  
 0504 660  
 0504 661  
 0504 662  
 0504 663  
 0504 664  
 0504 665 ----  
 0504 666  
 0504 667 DCL\$EOJ:::  
 0504 668 DCLSABORT:::  
 F9B6 CA 9E 0504 669 MOVAB WRK\_G\_RESULT(R10),- :RESET RESULT ARRAY PTR  
 BA AA 0508 670 WRK\_L\_RSLNXT(R10)  
 52 F492 CA 9E 050A 671 MOVAB WRK\_G\_BUFFER(R10),R2 :GET ADDRESS OF EXPANSION BUFFER  
 F486 CA 52 00 050F 672 MOVL R2,WRK\_L\_EXPANDPTR(R10) :RESET EXPANSION PTR  
 62 4F474F4C 8F 00 0514 673 MOVL #^A'LOGO', (R2) :STORE VERB INTO BUFFER  
 51 04 00 051B 674 MOVL #4,R1 :SET LENGTH OF VERB  
 E2 AA 00000000'EF 9E 051E 675 MOVAB LOGOUT,WRK\_L\_IMAGE(R10) :SET ROUTINE IN CASE NOT FOUND  
 58 00000000'GF 00 052A 676 SETBIT WRK\_V\_CLIRTN,WRK\_W\_FLAGS2(R10)  
 FACC' 30 0531 677 MOVL G^CTL\$AG\_CLITABLE,R8 :GET ADDRESS OF DATA BASE VECTOR  
 FAC9' 30 0534 678 BSBW DCLSSEARCH\_VERB :LOCATE COMMAND DEFINITION FOR  
 FD96 31 0537 680 BSBW DCL\$GENEOL :LOGOUT CLI INTERFACE (IGNORE ERRORS)  
 BRW BUILD\_IMAGE :CREATE AN EOL TOKEN  
 :PROCESS LOGOUT COMMAND

053A 683 .SBTTL EOD/DECK COMMANDS  
053A 684 :+ DCL\$EOD/DCL\$DECK  
053A 685 :+  
053A 686 :+  
053A 687 :+ THESE COMMANDS ARE PROCESSED BY RMS AND SHOULD NOT BE SEEN BY THE CLI.  
053A 688 :+ RMS PASSES THEM THROUGH IF THE SYNTAX OR VALUE IS BAD. ONE CASE IS  
053A 689 :+ NOT TREATED AS AN ERROR, THAT IS "EOD" ALONE ON A LINE. THIS COULD  
053A 690 :+ HAPPEN IF THE STATEMENT WAS ENTERED WITHOUT AN IMAGE RUN.  
053A 691 :+  
053A 692 :+  
053A 693 DCL\$EOD:  
FAC3' 30 053A 694 BSBW DCLSSETNBLK :+ LOOK FOR NEXT CHARATER IN LINE  
08 12 053D 695 BNEQ DCLSDECK :+ BR IF NOT A NULL LINE  
053F 696 STATUS NORMAL :+ SET OK STATUS  
05 0546 697 RSB :+  
0547 698 :+  
0547 699 DCLSDECK:  
05 0547 700 STATUS IVVALU :+ ASSUME ERROR WAS INVALID VALUE  
05 054E 701 RSB :+  
;

054F 703 .SBTTL CHECK FOR CONTROL Y/C AST PENDING  
054F 704 ::+ DCL\$CHECK\_AST - CHECK FOR CONTROL Y/C AST PENDING  
054F 705 : THIS ROUTINE IS CALLED TO CHECK FOR A PENDING CONTROL Y/C AST.  
054F 706 :  
054F 707 :  
054F 708 :  
054F 709 :  
054F 710 :  
054F 711 :  
054F 712 :  
054F 713 :  
054F 714 :  
054F 715 : IF A CONTROL Y/C AST IS PENDING, THEN CONTROL IS TRANSFERED TO THE COMMAND  
054F 716 : RESTART ENTRY POINT. OTHERWISE CONTROL IS RETURNED TO THE CALLER.  
054F 717 :  
054F 718 :  
054F 719 : ENABL LSB  
054F 720 DCL\$CHECK\_AST:: :CHECK FOR PENDING CONTROL Y/C AST  
054F 721 BBC #PRC\_V\_CNTRLY,PRC\_W\_FLAGS(R11),50\$ ;IF CLR, NO CONTROL Y/C REQUEST  
054F 722 BRW DCL\$RESTART ;  
2B 68 AB 01 E1 0556 722 10\$:

0557 724 .SBTTL ENABLE/DISABLE CONTROL Y/C AST'S  
 0557 725  
 0557 726 DCLSDISABLE - DISABLE CONTROL Y/C AST'S  
 0557 727  
 0557 728  
 0557 729  
 0557 730 THIS ROUTINE IS CALLED TO DISABLE/ENABLE CONTROL Y/C AST'S.  
 0557 731  
 0557 732  
 0557 733  
 0557 734  
 0557 735  
 0557 736  
 0557 737  
 0557 738  
 0557 739  
 0557 740  
 0557 741  
 0557 742  
 0557 743 DCLSDISABLE::  
 04 AE 68 AB 6E DD 0557 744 PUSHL (SP) :DISABLE CONTROL Y/C AST'S  
 13 04 AE 02 9E 16 E0 0559 745 MOVZWL PRC\_W\_FLAGS(R11),4(SP) :COPY RETURN ADDRESS  
 05 68 AB 08 0562 0564 746 SETBIT PRC\_V\_DISABL,PRC\_W\_FLAGS(R11) :SAVE PREVIOUS DISABLE STATE  
 5C AB D5 0569 747 JSB 0(SP)? :DISABLE CONTROL Y/C AST'S  
 DC 68 AB 01 E0 0571 748 BBS #PRC\_V\_DISABL,4(SP),40\$ :CALL THE CALLER BACK  
 6E 8ED0 057C 0578 749 BBS #PRC\_V\_YLEVEL,PRC\_W\_FLAGS(R11),20\$ :IF SET, AT CONTROL Y/C LEVEL  
 05 057F 750 TSTL PRC\_E\_INDEPTH(R11) :INDIRECT LEVEL ZERO?  
 0580 751 BNEQ 30\$ :IF NEQ NO  
 0580 752 20\$: BBS #PRC\_V\_CNTRLY,PRC\_W\_FLAGS(R11),10\$ :IF SET, CONTROL Y/C REQUEST  
 0580 753 30\$: CLRBIT PRC\_V\_DISABL,PRC\_W\_FLAGS(R11) :ENABLE CONTROL Y/C AST'S  
 0580 754 40\$: POPL (SP) :REMOVE PREVIOUS STATE FROM STACK  
 0580 755 50\$: RSB  
 0580 756 .DSABL LSB

0580 758 .SBTTL FLUSH COMMAND BUFFER  
 0580 759 ::+  
 0580 760 :: DCLSFLUSH - FLUSH COMMAND BUFFER  
 0580 761 ::  
 0580 762 :: THIS ROUTINE IS CALLED TO READ CHARACTERS FROM THE COMMAND BUFFER UNTIL AN  
 0580 763 :: END OF LINE IS ENCOUNTERED.  
 0580 764 ::  
 0580 765 :: INPUTS:  
 0580 766 ::  
 0580 767 :: NONE.  
 0580 768 ::  
 0580 769 ::  
 0580 770 ::  
 0580 771 :: OUTPUTS:  
 0580 772 :: INDIRECT FILE RECOGNITION IS DISABLED AND CHARACTERS ARE READ FROM THE  
 0580 773 :: INPUT BUFFER UNTIL AN END OF LINE IS ENCOUNTERED.  
 0580 774 ::-  
 0580 775 DCLSFLUSH:: :FLUSH COMMAND BUFFER  
 0580 776 BBS #PRC\_V\_YLEVEL,PRC\_W\_FLAGS(R11),20\$ ; IF AT "Y" LEVEL DON'T FLUSH  
 0580 777 MOVL PRC\_C\_INDINPRAB(R11),R0 ; GET CURRENT INPUT RAB  
 0580 778 BBS #DEV\$0 TRM\_RABSL\_CTX(R0),20\$ ; LIKEWISE FOR TERMINALS  
 0580 779 MOVAB WRK\_G\_BUFFER(R10),WRK\_L\_EXPANDPTR(R10) ;SET BUFFER TO RECEIVE ANY CO  
 0580 780 SETBIT PRC\_V\_IND,PRC\_W\_FLAGS(R11) ;DISABLE INDIRECT FILE RECOGNITION  
 0580 781 SETBIT PRC\_V\_FLUSH,PRC\_B\_FLAGS2(R11) ;SET FLUSH IN PROGRESS  
 0580 782 10\$: BSBW DCLSGETCHAR :GET CHARACTER FROM INPUT BUFFER  
 0580 783 BNEQ 10\$ :IF NEQ NOT END OF LINE  
 0580 784 CLRBIT PRC\_V\_FLUSH,PRC\_B\_FLAGS2(R11) ;RESET FLUSH IN PROGRESS FLAG  
 0580 785 CLRBIT PRC\_V\_IND,PRC\_W\_FLAGS(R11) ;ENABLE INDIRECT FILE RECOGNITION  
 0580 786 20\$: RSB ;

29 68 AB 0B E0  
 50 14 AB D0  
 20 18 A0 02 E0  
 F486 CA F492 CA 9E

FA5E' FB

30 059F  
 12 05A2  
 05A4  
 05AA  
 05 05AE

782 10\$: 783  
 784  
 785  
 786 20\$:

		05AF	788	.SBTTL PROCESS FOREIGN COMMAND	
		05AF	789	---	
		05AF	790	PROCFORN - GENERATE RESULT DESCRIPTOR FOR FOREIGN COMMAND LINE	
		05AF	791	PROCESS FOREIGN COMMAND AND BUILD TOKEN DESCRIPTOR	
		05AF	792	FOR THE REST OF THE LINE AFTER THE VERB.	
		05AF	793	INPUTS:	
		05AF	794	NONE	
		05AF	795	OUTPUTS:	
		05AF	796	A TOKEN DESCRIPTOR IS GENERATED FOR THE REST OF THE LINE AND	
		05AF	797	A TOKEN DESCRIPTOR IS GENERATED FOR THE END OF LINE.	
		05AF	798	---	
		05AF	799	PROCFORN:	
		05AF	800	PROCESS FOREIGN COMMAND	
		05AF	801	:DISABLE INDIRECT FILE INTERPRETATION	
		05AF	802	:PEEK AT NEXT CHARACTER IN INPUT BUFFER	
		05AF	803	:BLANK?	
		05AF	804	:BRANCH IF NO LEADING BLANK	
		05AF	805	:MOVE BLANK BEFORE MARK	
		05AF	806	:MARK POSITION AFTER FILESPEC & BLANK	
		05AF	807	:MOVE CHARACTER TO COMMAND BUFFER	
		05AF	808	:UNTIL END OF LINE	
		05AF	809	:BACKUP OVER EOL CHARACTER	
		05AF	810	:GET DESCRIPTOR OF REST OF LINE	
20	FA4A'	30	05B3	5\$:	:COPY DESCRIPTOR
	50	91	05B6	10\$:	:CLEAR ITEM NUMBER
	03	12	05B9	105:	:CLEAR FLAGS
	FA42'	30	05B8	105:	:CLEAR ENTITY BLOCK ADDRESS
	FA3F'	30	05BE	105:	:SET ITEM TYPE TO PARAMETER
	FA3C'	30	05C1	105:	:GENERATE RESULT PARSE TABLE DESCRIPTOR
	FB	12	05C4	105:	:SET ITEM TYPE TO END OF LINE
	F486	CA	05C6	105:	:SET LENGTH OF ITEM
	FA33'	30	05CA	105:	:SET STARTING ADDRESS OF ITEM
	57	51	05CD	105:	:GENERATE RESULT PARSE DESCRIPTOR
	54	D4	05D0	105:	:ENABLE INDIRECT FILE INTERPRETATION
	56	D4	05D2	105:	:
	59	D4	05D4	105:	
	55	03	9A	05D6	
	FA24'	30	05D9	105:	
	55	04	9A	05DC	
	57	01	9A	05DF	
58	F486	CA	05E2	105:	
	FA16'	30	05E7	105:	
	05	05EA	105:		
	05EE	05EF	105:		
			821	MOVZBL	#PTR_K_PARAMETR,R5
			822	BSBW	DCL\$GENDESCR
			823	MOVZBL	#PTR_K_ENDLINE,R5
			824	MOVZBL	#1,R7
			825	MOVL	WRK_L_EXPANDPTR(R10),R8
			826	BSBW	DCL\$GENDESCR
			827	CLRBIT	PRC_V_IND,PRC_W_FLAGS(R11)
			828	RSB	:ENABLE INDIRECT FILE INTERPRETATION
			829		:

05EF	831
05EF	832
05EF	833
05EF	834
05EF	835
05EF	836
05EF	837
05EF	838
05EF	839
05EF	840
05EF	841
05EF	842
05EF	843
05EF	844
05EF	845
05EF	846
05EF	847
05EF	848
05EF	849

.SBTTL GET INTERNAL ROUTINE INDEX

---  
DCLSLOCATE\_INTERNAL - GET INDEX FROM INTERNAL ROUTINE TABLE

THIS ROUTINE SEARCHES THE TABLE OF INTERNAL ROUTINES FOR A SPECIFIED STRING.  
IF FOUND, THE INDEX OF THAT STRING IS RETURNED.

INPUTS:

R1/R2 = ROUTINE NAME DESCRIPTOR

OUTPUTS:

R0 = STATUS  
R3 = INDEX NUMBER  
R4 IS DESTROYED

---  
DCLSLOCATE INTERNAL:

		7E	51	7D	05EF	850	MOVQ	R1,-(SP)	: SAVE R1/R2
		7E	62	7D	05F2	851	MOVQ	(R2),-(SP)	: GET ROUTINE NAME
		6E	80	8F	05F5	852	BISB	#^X80,(SP)	: SET HIGH ORDER BIT TO FORCE ROUTINE MATCH
		08	51	D1	05F9	853	CMPL	R1,#8	: ROUTINE LENGTH GREATER THAN 8?
		54	000001B0	8F	0601	854	BLEQ	SS	: BRANCH IF NOT
00000007'EF	54	6E	51	39	0608	855	MOVL	#8,R1	: SET LENGTH EQUAL 8
			17	12	0611	856	5B:	MOVL #ROUTINES,LEN,R4	: SET TABLE LENGTH
		52	07	CA	0613	857	MATCHC	R1,(SP),R4,ROUTINES	: FIND SPECIFIED ROUTINE
		52	08	CO	0616	858	BNEQ	20\$	: BRANCH IF NO MATCH
		54	52	C2	0619	859	BICL	#7,R2	: ROUND REMAINING LENGTH UP
		50	FD	78	061C	860	ADDL	#8,R2	: CALCULATE RELATIVE POSITION OF MATCH
		50	01	D0	0621	861	SUBL	R2,R4	: DIVIDE BY 8 TO GET INDEX INTO R3
		51	BE	7C	0624	862	ASHL	#-3,R4,R3	: SET SUCCESS
					0626	863	MOVL	#1,R0	: RESTORE STACK
			05	05	0629	864	10\$:	CLRQ (SP)+	: RESTORE R1/R2
					062A	865	MOVL (SP)+,R1		
					0631	866	RSB		
50	00038912	8F	D0	062A	867				
		F1	11	0631	868	20\$:	MOVL #CLIS_INVROUT,R0	: SET INVALID ROUTINE STATUS	
				0633	869		BRB	10\$	: RETURN
				0633	870				
				0633	871				

.END

SINTIMAGES	= 000000B6	DCLSGETOKEN	*****	X	03
SNAM1S	= 435349C4	DCLSGOTO	*****	X	03
SNAM2S	= 454E4F4F	DCLSHYPHEN	*****	X	03
BUILD_IMAGE	= 000002D0 R	DCLSIF	*****	X	03
CLISK_VERB FORE	***** X	DCLSINPUT	*****	X	03
CLIS_INVROOT	= 00038912	DCLSINQUIRE	*****	X	03
CLIS_IVVALU	= 00038088	DCLSLOCATE_INTERNAL	000005EF R		03
CLIS_NOCOMD	= 00038080	DCLSLOCKED_STATE	*****	X	03
CLIS_NOLBLS	= 000381E0	DCLSLOGOUT	*****	X	03
CLIS_NORMAL	= 00030001	DCLSMARK	*****	X	03
CMDVERB	= 00000276 R	DCLSMARKEDTOKEN	*****	X	03
CMD_V_FOREIGN	= 00000002	DCLSMCR	*****	X	03
CMD_V_IMMED	= 00000003	DCLSMOVCHAR	*****	X	03
CMD_W_FLAGS	= 00000004	DCLSON	*****	X	03
COMMAND	= 000001F1 R	DCLSOPEN	*****	X	03
CTL\$AG CLITABLE	***** X	DCLSPARSE_COMMAND	*****	X	03
DCLSABORT	= 00000504 RG	DCLSPROCFILE	*****	X	03
DCLSALLDEACMD	***** X	DCLSREAD	*****	X	03
DCLSALLOCATE	***** X	DCLSRECALL	*****	X	03
DCLSALLOC LABEL	***** X	DCLSRESTART	00000000 RG		03
DCLSASSIGN	***** X	DCLSRUN	*****	X	03
DCLSATTACH	***** X	DCLSSEARCH_VERB	*****	X	03
DCLSBACKUPCHAR	***** X	DCLSSETCHAR	*****	X	03
DCLSBACKUPMOVE	***** X	DCLSSETCTLY	*****	X	03
DCLSCANCEL	***** X	DCLSSETDEFLT	*****	X	03
DCLSCHARERROR	***** X	DCLSSETFLUSH	*****	X	03
DCLSCHECK_AST	= 0000054F RG	DCLSSETKEY	*****	X	03
DCLSCLOSE	***** X	DCLSSETNBLK	*****	X	03
DCLSCMDSTART	= 00000011 RG	DCLSSETON	*****	X	03
DCLSCONNECT	***** X	DCLSSETPROMPT	*****	X	03
DCLSCONTINUE	***** X	DCLSSETPROT	*****	X	03
DCLSCRETABLE	***** X	DCLSSETUIC	*****	X	03
DCLSDEALGOTO	***** X	DCLSSETVERIFY	*****	X	03
DCLSDEALLOCAT	***** X	DCLSSET STATUS	*****	X	03
DCLSDEASSIGN	***** X	DCLS\$SHOWDEF	*****	X	03
DCLSDEBUG	***** X	DCLS\$SHOWKEY	*****	X	03
DCLSDECK	= 00000547 RG	DCLS\$SHOWPROT	*****	X	03
DCLSDEFINE	***** X	DCLS\$SHOWQUOTA	*****	X	03
DCLSDEFKEY	***** X	DCLS\$SHOWSTAT	*****	X	03
DCLSDELKEY	***** X	DCLS\$SHOWSYMBL	*****	X	03
DCLSDELSYM	***** X	DCLS\$SHOWTIME	*****	X	03
DCLSDEPOSIT	***** X	DCLS\$SHOWTRAN	*****	X	03
DCLSDISABLE	***** X	DCLS\$SORT TOKENS	*****	X	03
DCLSDisconnect	= 00000557 RG	DCLS\$SPAWN	*****	X	03
DCL\$EO0	***** X	DCLS\$SPECIAL	*****	X	03
DCL\$EOJ	= 0000053A RG	DCLS\$STOP	*****	X	03
DCL\$EQUATE	***** X	DCLS\$SUBASSIGN	*****	X	03
DCL\$ERRORMSG	***** X	DCLS\$SYM STRING	*****	X	03
DCL\$EVALUATE	***** X	DCLS\$WAIT	*****	X	03
DCL\$EXAMINE	***** X	DCLS\$WRITE	*****	X	03
DCL\$EXIT	***** X	DEV\$V_RND	*****	X	03
DCL\$EXTIMAGE	***** X	DEV\$V_TRM	*****	X	03
DCLSFLUSH	= 00000580 RG	ERROR	= 0000001C		
DCLSFORCEEXIT	***** X	ERROR EXIT	= 00000002		
DCLSGENDESCR	***** X	EXEC\$XT	0000282 R	03	
DCLSGENEOL	***** X	FORVERB	0000337 R	03	
DCLSGETCHAR	***** X	IMAGECASE	00002F0 R	03	
			0000255 R	03	
			0000349 R	03	

```

IMG_K_ALLOCATE          = 00000080
IMG_K_ASSIGN             = 00000081
IMG_K_ATTACH             = 000000A9
IMG_K_CANCEL             = 000000AB
IMG_K_CLOSE              = 00000082
IMG_K_CONNECT            = 00000084
IMG_K_CONTINUE           = 00000083
IMG_K_CREATABLE          = 000000B3
IMG_K_DEALLOCAT          = 00000084
IMG_K_DEASSIGN           = 00000085
IMG_K_DEBUG              = 00000086
IMG_K_DECK               = 00000087
IMG_K_DEFINE              = 00000088
IMG_K_DEFKEY              = 000000AE
IMG_K_DELKEY              = 000000B0
IMG_K_DELSYM              = 000000BA
IMG_K_DEPOSIT             = 000000B9
IMG_K_DISCONNECT          = 000000B5
IMG_K_EOD                 = 0000008B
IMG_K_EOJ                 = 000000AA
IMG_K_EXAMINE             = 0000008C
IMG_K_EXIT                 = 0000008D
IMG_K_EXTIMAGE            = 000000BE
IMG_K_GOTO                 = 000000BF
IMG_K_IF                  = 00000090
IMG_K_INQUIRE              = 00000091
IMG_K_LOGOUT              = 00000092
IMG_K_MCR                 = 00000093
IMG_K_ON                  = 00000094
IMG_K_OPEN                 = 00000095
IMG_K_READ                 = 00000096
IMG_K_RECALL              = 000000AD
IMG_K_RUN                  = 00000097
IMG_K_SETCTLY              = 00000098
IMG_K_SETDEFAULT           = 00000099
IMG_K_SETFLUSH              = 000000B2
IMG_K_SETKEY              = 000000B1
IMG_K_SETON                 = 0000009A
IMG_K_SETPROMPT            = 000000AC
IMG_K_SETPROT              = 0000009B
IMG_K_SETUIC                 = 0000009C
IMG_K_SETVERIFY             = 0000009D
IMG_K_SHOWDEF              = 0000009E
IMG_K_SHOWKEY              = 000000AF
IMG_K_SHOWPROT             = 0000009F
IMG_K_SHOWQUOTA            = 000000A0
IMG_K_SHOWSTAT             = 000000A1
IMG_K_SHOWSYMBL            = 000000A2
IMG_K_SHOWTIME              = 000000A3
IMG_K_SHOWTRAN              = 000000A4
IMG_K_SPAWN                 = 000000A8
IMG_K_STOP                  = 000000A5
IMG_K_WAIT                  = 000000A6
IMG_K_WRITE                 = 000000A7
IMMED                     = 000002FA R 03
LOGOUT                     = 00000000 R 02
NOCOMD                     = 0000027B R 03

```

```

NULLNODE
PRCVERB
PRC_B_CONTINUE
PRC_B_DEFRADIX
PRC_B_EXMDEPMOD
PRC_B_EXMDEPWID
PRC_B_EXONLYL
PRC_B_FLAGS2
PRC_B_IMGFLAG
PRC_B_OUTFLAGS
PRC_B_PROMPTLEN
PRC_C_LENGTH
PRC_G_COMMANDS
PRC_G_PROMPT
PRC_K_LENGTH
PRC_L_CURRKEY
PRC_L_EXMDEPADR
PRC_L_EXTARG
PRC_L_EXTBLK
PRC_L_EXTCOD
PRC_L_EXTHND
PRC_L_EXTPRM
PRC_L_IDFLNK
PRC_L_IMGACTSTS
PRC_L_INDCLOCK
PRC_L_INDEPTH
PRC_L_INDFA
PRC_L_INDINPRAB
PRC_L_INDOUTRAB
PRC_L_INPRAB
PRC_L_LASTKEY
PRC_L_LSTSTATUS
PRC_L_ONCTLY
PRC_L_ONERROR
PRC_L_OUTOFBAND
PRC_L_OUTRAB
PRC_L_OUTRABCTX
PRC_L_PPFLIST
PRC_L_RECALLPTR
PRC_L_RESTART
PRC_L_SAVAP
PRC_L_SAVFP
PRC_L_SEVERITY
PRC_L_SPWN
PRC_L_STACKLM
PRC_L_STACKPT
PRC_L_STATUS
PRC_L_STS
PRC_L_STV
PRC_L_SYMBOL
PRC_L_TMBX
PRC_L_TRMLIST
PRC_M_DISABL
PRC_M_FLUSH
PRC_M_IND
PRC_M_RUNDEF
PRC_Q_ALLOCREG

```

COMMAND  
Symbol table

- PROCESS NEXT COMMAND

C 9

15-SEP-1984 23:40:03 VAX/VMS Macro V04-00  
4-SEP-1984 23:39:43 [DCL.SRC]COMMAND.MAR;1

Page 25  
(12)

PRC_Q_COMMAND	000000E0	SYM_T_SYMBOL	0000000C
PRC_Q_FLUSHTIME	000000D0	SYM_W_SIZE	00000008
PRC_Q_GLOBAL	00000028	WRK_B_CMDOPT	FFFFFC3
PRC_Q_IMAGENAME	000000D8	WRK_B_MAXPARM	FFFFFD0
PRC_Q_KEYPAD	00000040	WRK_B_MINPARM	FFFFFD1
PRC_Q_LABEL	00000030	WRK_B_PARMCNT	FFFFFC6
PRC_Q_LOCAL	00000038	WRK_B_PARMSUM	FFFFFCF
PRC_Q_SAVEPRIV	000000E8	WRK_B_RECALLCNT	FFFFFC5
PRC_T_OUTDVI	0000011C	WRK_B_VALLEV	FFFFFC4
PRC_V_CHAIN	= 00000001	WRK_B_VERBTYP	FFFFFC2
PRC_V_CMD	= 00000000	WRK_C_LENGTH	FFFFF486
PRC_V_CNTRLY	= 00000001	WRK_C_SCRSTSIZ	= 00000200
PRC_V_DISABL	= 00000002	WRK_G_BUFFER	FFFFF492
PRC_V_FLUSH	= 00000006	WRK_G_INPBUF	FFFFF896
PRC_V_GOTO	= 00000004	WRK_G_RESULT	FFFF9B6
PRC_V_HANGUP	= 0000000C	WRK_K_LENGTH	FFFFF486
PRC_V_IND	= 00000005	WRK_L_CHARPTR	FFFFF48E
PRC_V_MODE	= 00000006	WRK_L_DISALLOW	FFFFFE6
PRC_V_PRIV	= 00000004	WRK_L_ERRORRTN	FFFF9AE
PRC_V_RUNDEF	= 00000002	WRK_L_EXPANDPTR	FFFFF486
PRC_V_YLEVEL	= 0000000B	WRK_L_IMAGE	FFFFFE2
PRC_W_ASTIOSB	000000C6	WRK_L_MARKPTR	FFFFF48A
PRC_W_ASTRETN	000000C8	WRK_L_PAROUT	FFFFFD2
PRC_W_ASTSTATUS	000000C4	WRK_L_PMPADDR	FFFFF9A2
PRC_W_ATTMBX	0000007A	WRK_L_PROMPTRN	FFFFF9A6
PRC_W_FLAGS	00000068	WRK_L_PROPTR	FFFFFC6
PRC_W_INPCHAN	00000064	WRK_L_QUABLK	FFFFFC9A
PRC_W_ONLEVEL	0000006A	WRK_L_READRTN	FFFFF9AA
PRC_W_OUTIFI	00000114	WRK_L_RECALLPTR	FFFFFEA
PRC_W_OUTISI	00000116	WRK_L_RSLEND	FFFFFB6
PRC_W_OUTMBXCHN	000000CA	WRK_L_RSLNXT	FFFFFB8
PRC_W_OUTMBXREF	000000CE	WRK_L_SAVAP	FFFFFFF8
PRC_W_OUTMBXSIZ	000000CC	WRK_L_SAVFP	FFFFFFFC
PRC_W_PMPCTRL	000000F1	WRK_L_SAVSP	FFFFFFF4
PRC_W_WAITIOSB	00000066	WRK_L_SIGNALRTN	FFFFFD6
PROCFORN	000005AF	WRK_L_SPECRTN	FFFFF9B2
PSLSV_CURMOD	= 00000018	WRK_L_TAB_VEC	FFFFFD8
PTR_B_LEVEL	00000004	WRK_L_VERB	FFFFF8BE
PTR_B_NUMBER	00000005	WRK_V_CLIRTN	= 00000000
PTR_B_PARMCNT	00000006	WRK_V_COMMAND	= 00000001
PTR_B_VALUE	00000000	WRK_W_FLAGS	FFFFFFF0
PTR_C_LENGTH	0000000C	WRK_W_FLAGS2	FFFFFFF2
PTR_K_ENDLINE	= 00000004	WRK_W_IMGCHAN	FFFFF8EE
PTR_K_LENGTH	0000000C	WRK_W_PMPLEN	FFFFF99E
PTR_K_PARAMTR	= 00000003	XXX\$ALLOCATE	000003C0 R 03
PTR_L_DESCR	00000000	XXX\$ASSIGN	000003C6 R 03
PTR_L_ENTITY	00000008	XXX\$ATTACH	000004B6 R 03
RABSL_CTX	= 00000018	XXX\$CANCEL	000004C2 R 03
ROUTINES	00000007	XXX\$CLOSE	000003CC R 03
ROUTINES_LEN	= 000001B0	XXX\$CONNECT	000004F8 R 03
SSS_ILLSER	*****	XXX\$CONTINUE	000003D2 R 03
SYMBOL	000001DB	XXX\$CRETABLE	000004F2 R 03
SYM_B_FLAGS	0000000B	XXX\$DEALLOCAT	000003D8 R 03
SYM_B_NONUNIQUE	0000000B	XXX\$DEASSIGN	000003DE R 03
SYM_B_TYPE	0000000A	XXX\$DEBUG	000003E4 R 03
SYM_L_BL	00000004	XXX\$DECK	000003EA R 03
SYM_L_FL	00000000	XXX\$DEFINE	000003F0 R 03

## COMMAND Symbol table

- PROCESS NEXT COMMAND

D 9

15-SEP-1984 23:40:03 VAX/VMS Macro V04-00  
4-SEP-1984 23:39:43 [DCL.SRC]COMMAND.MAR;1

Page 26  
(12)

XXX\$DEFKEY	000004D4	R
XXX\$DELKEY	000004E0	R
XXX\$DELSYM	000003FC	R
XXX\$DEPOSIT	000003F6	R
XXX\$DISCONNECT	000004FE	R
XXX\$EOOD	00000402	R
XXX\$EOJ	000004BC	R
XXX\$EXAMINE	00000408	R
XXX\$EXIT	0000040E	R
XXX\$EXTIMAGE	00000414	R
XXX\$GOTO	0000041A	R
XXX\$IF	00000420	R
XXX\$INQUIRE	00000426	R
XXX\$LOGOUT	0000042C	R
XXX\$MCR	00000432	R
XXX\$ON	00000438	R
XXX\$OPEN	0000043E	R
XXX\$READ	00000444	R
XXX\$RECALL	000004CE	R
XXX\$RUN	0000044A	R
XXX\$SETCTLY	00000450	R
XXX\$SETDEFAULT	00000456	R
XXX\$SETFLUSH	000004EC	R
XXX\$SETKEY	000004E6	R
XXX\$SETON	0000045C	R
XXX\$SETPROMPT	000004C8	R
XXX\$SETPROT	00000462	R
XXX\$SETUIC	00000468	R
XXX\$SETVERIFY	0000046E	R
XXX\$SHOWDEF	00000474	R
XXX\$SHOWKEY	000004DA	R
XXX\$SHOWPROT	0000047A	R
XXX\$SHOWQUOTA	000004B0	R
XXX\$SHOWSTAT	000004B6	R
XXX\$SHOWSYMBL	000004BC	R
XXX\$SHOWTIME	00000492	R
XXX\$SHOWTRAN	00000498	R
XXX\$SPAWN	000004B0	R
XXX\$STOP	0000049E	R
XXX\$WAIT	000004A4	R
XXX\$WRITE	000004AA	R

! Psect synopsis !

**PSECT name**

+-----+  
! Performance indicators !  
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	18	00:00:00.04	00:00:01.49
Command processing	97	00:00:00.64	00:00:06.46
Pass 1	361	00:00:13.86	00:00:43.10
Symbol table sort	0	00:00:01.58	00:00:04.96
Pass 2	181	00:00:03.27	00:00:09.98
Symbol table output	47	00:00:00.32	00:00:01.20
Psect synopsis output	4	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	708	00:00:19.75	00:01:07.23

The working set limit was 1200 pages.

87010 bytes (170 pages) of virtual memory were used to buffer the intermediate code.

There were 60 pages of symbol table space allocated to hold 1080 non-local and 49 local symbols.

871 source lines were read in Pass 1, producing 26 object records in Pass 2.

43 pages of virtual memory were used to define 27 macros.

+-----+  
! Macro library statistics !  
+-----+

Macro library name	Macros defined
\$255SDUA28:[SYSLIB]SYSBLDMLB.MLB;1	0
\$255SDUA28:[DCL.OBJ]DCL.MLB;1	13
\$255SDUA28:[SYS.OBJ]LIB.MLB;1	0
\$255SDUA28:[SYSLIB]STARLET.MLB;2	6
TOTALS (all libraries)	19

1101 GETS were required to define 19 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LISS:COMMAND/OBJ=OBJ\$:COMMAND MSRC\$:COMMAND/UPDATE=(ENHS:COMMAND)+EXECMLS/LIB+LIBS:DCL/LIB+SYSSLIBRARY:SYSBLDMLB/LIB

0069 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

INTDEF  
SDL

## COMMAND LIS

DESCRIVAL  
ETIS

CLIMAC  
MAR

## CONVERT LIS

CONNECT  
US

CHARMANIP  
LIS

CLIMSG  
LIS

DCLPARS  
LIS

EXAMDEP  
LTS

INTIMAGES  
MOR

DOCKSTAR  
LTS